

;login:

January / February 1993 Vol. 18, No. 1

From the Editor's Desk

Happy New Year!

1993 is surely going to be an interesting year. In the United States, we have a government in which the law making body and the President are both of the same political party for the first time in a long while. Things are going to change – I have no idea how, but they will change.

USENIX is trying a number of new ideas at the San Diego conference (several parallel tracks, to name one). We'll have the results of that experiment soon.

The computer world is seeing three volt logic being developed at an incredible rate (to meet the demand for portable computing). New operating systems that enable network connections from mobile computers are emerging. Pen-based operating systems are poised for a big strike this year. Windows NT will bow in quantity – and we'll find out how much it costs and whether the market wants to upgrade their DOS operating system.

I think we'll see much greater use of interpreters this year and in the near future: there's no reason to compile many kinds of specifications when it's just as easy to interpret some high level representation (e.g., of a menu layout).

I also think we'll see UNIX moving more and more into the high levels of computing in the corporate MIS world. I don't know what will happen where the MIS department interfaces with department level computing, though.

It sure looks like the Internet will continue to grow in a big way. ISDN is becoming more and more widely available – that should yield some interesting application.

All in all, 1993 should be fascinating. I hope it's good for you!

RK

The closing date for submissions to the next issue of ;login: is February 22, 1993.

Association News

President's Letter.....	3
<i>Stephen C. Johnson</i>	
UNIX Security Symposium Report	4
<i>Ed DeHart and Barbara Frazer</i>	
A Changing of the Guard	5
<i>Kirk McKusick</i>	
Board Meeting Summary	6
<i>Ellie Young</i>	

SAGE News

SAGE Elections Results	10
System Administration Tools	10
<i>Elizabeth Zwicky</i>	
SAGE: Tools.....	11
<i>Steve Simmons</i>	
Views	12
<i>Bjorn Satdeva</i>	

SAGE Book Reviews:

Help! The Art of Computer Technical Support.....	16
UNIX System Performance Tuning	17
SAGE: A Proposed New Model	18
<i>Peg Schafer</i>	

SAGE Call for Tools Reviews	22
-----------------------------------	----

Features

UNIX is Dead; Long Live UNIX	23
<i>Mike Devaney</i>	
Response to FileSystem Report	25
<i>Noemi Paciorek and Marc J. Teller</i>	
DOS Digs in Its Heels	26
<i>Bill Gallmeister</i>	
An Update on UNIX-Related Standards Activities.....	27
<i>Stephen Walli</i>	

Book Reviews

The Bookworm	47
<i>Peter H. Salus</i>	

Announcements

Summer 1993 Conference	49
Mach III Symposium	51
Microkernels II Symposium	53
SEDMS IV.....	55
UNIX Security IV Symposium	57
Mobile Computing Symposium	59
LISA '93	61
SANS II.....	63
Special Offer: <i>Computing Systems</i>	66
Publications Order Forms.....	68
Local User Groups	72
Calendar of Events.....	75



The UNIX and Advanced Computing Systems
Professional & Technical Association

General Information

login: is the official newsletter of the USENIX Association.

login: (ISSN 1044-6397) Volume 18, Number 1 (January/February 1993) is published bimonthly by the USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710. \$24 of each member's annual dues is for an annual subscription to *login*.. Subscriptions for non-members are \$50 per year. Second-class postage paid at Berkeley, CA and additional offices. POSTMASTER: Send address changes to *login*., USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710.

Contributions Solicited

You are encouraged to contribute articles, book reviews, and announcements to *login*.. Send them via email to login@usenix.org or through the postal system to the Association office. Send SAGE material to bigmac@erg.sri.com. The Association reserves the right to edit submitted material. Any reproduction of this newsletter in its entirety or in part requires the permission of the Association and the author(s).

Editorial Staff

Rob Kolstad, Editor <kolstad@usenix.org>
Ellie Young, Staff Editor <ellie@usenix.org>
Carolyn S. Carr, Managing Editor and Typesetter <carolyn@usenix.org>
Stephen Walli, Standards Report Editor <stephe@mks.com>
Bryan MacDonald, SAGE Editor <bigmac@erg.sri.com>
Michelle Dominijanni, Copy Editor

Membership and Publications

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710
Telephone: 510/528-8649
FAX: 510/548-5738
Email: <office@usenix.org>

Copyright © 1993 USENIX Association. The Association acknowledges all trade references made herein.

login: was produced on a SUN 3/50 with Framemaker 3.0 software provided by Frame Technology and printed on a QMS 860 donated by Quality Micro Systems. The newsletter is printed on recycled paper. ♻

Conferences & Symposia

Judith F. DesHarnais, Conference Coordinator
USENIX Conference Office
22672 Lambert Street, Suite 613
El Toro, CA 92630
Telephone: 714/588-8649
FAX: 714/588-9706
<conference@usenix.org>

Tutorials

Daniel V. Klein, Tutorial Coordinator
Telephone: 412/421-2332
<dvk@usenix.org>

Supporting Members

Digital Equipment Corporation
Frame Technology, Inc.
Matsushita Graphic Communication Systems
mt Xinu
Network Computing Devices, Inc.
Open Software Foundation
Quality Micro Systems
Sun Microsystems, Inc.
Sybase, Inc.
UNIX System Laboratories, Inc.
UUNET Technologies, Inc.

USENIX Association Board of Directors

If a member wishes to communicate directly with the USENIX Board of Directors, he or she may do so by writing to board@usenix.org

President:

Stephen C. Johnson <scj@usenix.org>

Vice President:

Michael D. O'Dell <mo@usenix.org>

Secretary:

Evi Nemeth <evi@usenix.org>

Treasurer:

Rick Adams <rick@usenix.org>

Directors:

Eric Allman <eric@usenix.org>
Tom Christiansen <tchrist@usenix.org>
Lori Grob <grob@usenix.org>
Barry Shein <bzs@usenix.org>

USENIX Association

Executive Director

Ellie Young <ellie@usenix.org>

President's Letter

Some of Your Best Friends are Marketeers...

by Stephen C. Johnson

<scj@usenix.org>

I sure hear a lot of putdowns of marketing people from the technical community. I hear a few put-downs (and I'm sure there are many more) of technical people from the marketing community. And yet the two functions are as clearly interconnected as tooth and gum. Most technical people would not eat without the timely intervention of a marketing person or two in their company; most marketing people do occasionally like to have a product or two to deliver (although the best marketeers try to rise above this shortcoming...).

Why is it so hard for techies to get along with marketeers? I think a lot has to do with a very fundamental question: what is Truth? As technical people, we are taught to honor truth above all things. While there is little classical science in "Computer Science," we still would like to think we are engaged in an objective activity. Certainly when doing quality assurance, debugging, or many programming tasks, an uncompromising willingness to look at the state of the program clearly, warts and all, is one of our biggest assets. We are so wedded to the truth that we even put BUGS sections on our man pages.

The marketing world turns on a different axis – the pleasure principle. If it feels good, people will buy it. Sometimes the Truth feels good, and everybody is happy. Most of the time, the whole Truth is a bit uncomfortable, a bit more than most people can feel good about.

For example, suppose when you bought a new car you were handed a four page sheet on which were listed all the places where the car failed to meet the highest automotive standards – all the pieces of shim, the screws in crooked, the scratches in the underbody. I think you'd enjoy driving less with this information; most of the problems will never affect you at all, and you'd probably even admit that most cars have numerous such problems, but when your nose is rubbed in it you don't feel good. We all know someone who, claiming to be truthful, tells us how much better we would look with a little liposuction and a face lift...

At its best, marketing lubricates the marketplace, giving potential customers information about products that may perfectly match their needs. Even in this case, the information is not the whole Truth; the glossy ads rarely include the release notes! But pure informational marketing is rare; if the customer can also be made to feel good about a product, they will be happier customers, tell their friends, etc. The problems arise when, in an urge to make the customers happy, the marketing information loses touch with reality. As one excellent marketeer told me, "Marketing is not constrained by natural law." All you need to do to sell something is to be able to think it, not build it; here lies the biggest source of tension between technical and marketing, reality and illusion.

I think this tension is healthy. The marketing people are the customers' representatives in the planning meetings and day to day operations of a company. And of course customers want a gazillion features tomorrow (this afternoon would be better), and they should be free (although they might go as high as five dollars). In companies where there is no marketing/technical tension, either the marketing people are out of touch with the customers, or the technical people are agreeing to too much. Or worst of all, marketing and technical people aren't communicating at all.

It is a lot easier for us not to communicate with marketing than it is to walk daily into meetings and try to work with the resulting tensions. And what better way to give ourselves permission not to communicate with someone than to put them down. On the marketing side, the tension is there too; here they see a wonderful opportunity to gain a customer for life if we could only deliver next year's product by Friday, and those technical people stubbornly refuse to do it!

The keys are respect and communication. And tolerating a bit of tension (think of it as a mental Nautilus machine; as you work on it, it gets easier). Don't give up the Truth, since ultimately the product won't work if you don't keep your eye on the ball. But if you also make room for the pleasure principle in your work life, who knows, you might have more fun on the job.

UNIX Security Symposium Report

Third USENIX UNIX Security Symposium Report

by Edward DeHart and Barbara Fraser

CERT Coordination Center

<ecd@cert.org> <byf@cert.org>

The Third USENIX UNIX Security Symposium was held September 14-16, 1992 in Baltimore, Maryland. Attendance this year was 286, an increase from previous symposiums. In addition, there was increased international participation, both in submissions and attendance.

This year's symposium was expanded to include a full day of tutorials. The tutorials given were "Network Security: The Kerberos Approach" presented by Dan Geer and Jon Rochlis, and CERT's "Internet Security for UNIX System Administrators." We were pleased that 200 of the 286 attendees participated in one of the tutorials.

Another addition to this year's event was the inclusion of an evening of Birds of a Feather sessions. These were heavily attended and one session on firewalls has continued as a mailing list.

The symposium was a success not only in the number of attendees, but more importantly, by the quality of the technical content of the presentations. Results of the attendees' surveys showed that 86% felt that the overall technical content was above average or better. We also received a large number of positive comments regarding the fact that many presentations provided information about practical security tools that are available for use today.

The keynote speaker of this year's symposium was Scott Charney of the U.S. Department of Justice. Mr. Charney spoke on the Justice Department's Computer Crime Initiative. His talk was enthusiastically received by attendees and lively discussion on several topics (including keystroke logging, liability issues, and investigative issues) followed his presentation.

In prior years, this symposium was organized with a single track. This year, there were so many quality submissions that we added a second track for the afternoon of the last day. Technical presentations covered the following general categories: war stories, TCP/IP network security, tools, applied research, and multi-level security.

In the past this symposium was held every other year. This year's symposium was so successful that we look forward to making this a popular annual event. The Fourth USENIX UNIX Security Symposium will be held October 4-7, 1993 at the Santa Clara Marriott, in Santa Clara, California. (See page 57 for the Call for Papers.)

The proceedings are available for purchase from USENIX (see page 68 for order information).

1 Changing of the Guard

A Changing of the Guard for the USENIX Institutional Representative to POSIX

by Kirk McKusick

Past President, USENIX

At its Fall meeting, the USENIX Board of Directors authorized funds for 1993 for an Institutional Representative (IR) to the IEEE Computer Society Technical Committee on Operating Systems (TCOS).

The USENIX IR has two basic responsibilities. First, to be an informed participant representing the USENIX membership in the POSIX activities. Being an informed participant requires attending POSIX meetings, reading the mailings, and discussing and soliciting input about the activities from technical experts and the USENIX membership.

The second task is to feed information about the POSIX activities back to the USENIX membership. This feedback is done through the snitch reports that appear after each POSIX meeting in `comp.std.unix` and in this newsletter. Through these reports, USENIX provides critical information to both its members and other interested individuals worldwide.

Peter Collinson has held the IR post for the past two years. While he has been active in the UNIX community for many years, Peter was new to the POSIX community. It took him relatively little time, however, to become immersed in the politics of POSIX and begin making significant contributions both at the meetings and in the subsequent reports to USENIX. Two years hence (and six trips annually from England to POSIX and USENIX meetings in the United States), Peter has indicated his desire to step down at the end of this year. Peter's contributions have been highly val-

ued, and I am sure he will be missed by the POSIX community. He brought a useful focus on "reality" at times when many folks were caught up in what is best characterized as "religious zeal."

Last summer, the USENIX Board of Directors formed a subcommittee with Peter Collinson's assistance to search for candidates. We are pleased to announce that the IR position has been offered to Jeffrey Haemer, of Canary Software, Inc., Boulder, Colorado, and he has accepted.

Jeff has been involved with standards and USENIX for many years. He served as the USENIX watchdog editor writing, coordinating, and editing articles about UNIX-related standards activities for USENIX. He has also attended POSIX meetings on and off since their inception. Jeff has lectured on standards, portability, internationalization, and open systems at shows and conferences that include USENIX, 6th Annual Berkeley Developer's Conference, Sun Expo, IEEE's Comp-Con, and the First and Second Gulf UNIX Conferences (in Kuwait, before and after the war).

Besides his duties at POSIX meetings, you will see him at USENIX conferences, where he will coordinate the Standard BOFs, discuss standards issues with our membership, recruit and instruct snitches, and work with the snitch editor (Stephen Walli) in publishing the reports.

Jeff will be attending the four IEEE POSIX meetings in 1993. He can be reached via electronic mail: jsh@canary.com or by phone: +1-303-494-0924. Welcome aboard, Jeff!

Board Meeting Summary

by **Ellie Young**
Executive Director

Below is a summary of the actions taken at the regular quarterly meeting of the USENIX Board of Directors held in Berkeley, CA on October 24 and 25, 1992.

Attendance: Rick Adams, Eric Allman, Tom Christiansen, Lori Grob, Steve Johnson, Kirk McKusick, Evi Nemeth, Mike O'Dell, Barry Shein, Ellie Young, Judy DesHarnais

Budget, Funding, and Fee Proposals

Budget. Young explained that while a small deficit had been budgeted for 1992, current projections revealed a substantial net. She said that while she would like to have a better way to predict the budget more accurately, the major factor affecting the predictions is the difficulty in correctly estimating attendance at the smaller conferences. A discussion ensued regarding conference and workshop attendance, ways to save in expenses, and factors that effect variable vs. fixed expenses.

O'Dell expressed concern that membership was flat. Shein felt this could be linked to lower conference attendance. Grob stressed the need to think about promotion, to have some goals concerning membership growth, and to use the database to extract information. Allman volunteered to take a look at the technical issues concerning the database.

Young explained that the 1993 preliminary draft budget contained conservative estimates for attendance at the conferences. Johnson went through the details and assumptions made on the cover report. It was decided to create a general promotional line item in the budget of \$20,000 to be overseen by the promotion committee. Young reported that the SAGE interim board had approved its budget.

Standards IR and Snitch Editor. The IR search committee had agreed on a candidate to replace Collinson and recommended Jeff Haemer as the next Institutional Representative. Funding for snitch and IR activities was approved.

ISO Monitor Representative. Walli resigned from his post of ISO Monitor to USENIX/EurOpen because of his work schedule. The proposal for

funding would be discussed by both organizations in the future.

Student Stipends. It was agreed to increase the amount for workshops from \$15,000 to \$20,000 for 1993.

USACO Proposal. It was agreed to allocate \$3,000 to fund the International Computing Olympiad (USACO) in order to foster computing at the pre-college level.

Membership Dues. It was agreed to raise corporate member dues by \$25 to \$325.

Conference Fees. It was decided to raise conference fees after January, 1993, as follows:

1 day of tutorials	\$275
2 days of tutorials	\$495
3 days tech sessions	\$295
2 days of tech sessions	\$275

Student fees would remain the same.

motd. It was decided to replace the current motd conference newsletter at the San Diego conference, with a one page schedule of events.

Honorary Member

It was agreed to make Lou Katz, one of the Association's founders and president for many years, an honorary lifetime member.

SAGE Report

Christiansen reported that most of the SAGE board attended the LISA conference; that 12 candidates came forward for 6 slots in the upcoming elections; and that there was a very successful candidate's forum. It was suggested that for the SAGE budget the items for Board Travel, Awards and Other be combined into a Misc. Other item and the value be set to \$5,000, with the incoming SAGE board being able to allocate these funds as they see fit.

Special Technical Group (STG) Document Committee Report

Johnson reported that the committee had made revisions to the STG portion of the document, but the Local Technical Group portion needed to worked on. Allman and Johnson volunteered to work on a proposal regarding the vote by email clauses in the USENIX Bylaws.

C++ Conference. It was reported that while attendance was somewhat less than the previous year's event, interest remained high. Young was beginning plans for having another one in Spring '94 with Doug Lea as program chair, and more promotion of this event would be needed.

Security Symposium. Young and DesHarnais reported that interest and technical content of the meeting was high. Ed DeHart did an excellent job as chair. Young was discussing with him plans for another event next year, and Bill Cheswick was interested in serving as program chair.

SANS II. It was agreed that SAGE/USENIX would co-sponsor (with FedUNIX) the Spring '93 Conference on Tools & Techniques for System Administration, Networking and Security per the enclosed proposal.

Site Selection for Summer '97. It was decided that we should investigate holding it in Chicago.

Mobile Computing Workshop Proposal. Shein explained that the attached proposal from Geer was a great topic, and it was approved.

Network Administration Workshop Proposal. It was felt that while the proposal from Chapman was an interesting topic, there were concerns that it might effect the LISA conference in November. It was suggested that the proposal be discussed further with the newly elected SAGE board.

Awards

O'Dell brought a sample of the glass sculpture, "The Flame" and its base, which would be the first USENIX Lifetime Achievement Award to recognize and celebrate singular contributions to the UNIX community in both intellectual achievement and service that are not recognized in any other forum. The committee proposed that the first award be made to CSRG.

UniForum, EurOpen & Cooperation with Other Groups

UniForum. Johnson explained that he and Young had been contacted by Mike Tilson, who is on their board and was a former director of USENIX. UniForum's board would like to have closer cooperation with USENIX. We would explore the possibility of perhaps holding an event alongside the UniForum Conference in the future, as well as cooperating in educational and standards activities.

EurOpen. It was agreed that we don't want a hierarchical relationship with the national groups of EurOpen, and that we might be able to assist them by providing tutorials and publications, and that we should look for specific opportunities and proposals in the future.

SUG. It was decided that USENIX is interested in discussing the possibility of making SUG an STG associated with USENIX, as there are some areas that we might have some synergy. Several of the Board and Executive Director would discuss this with SUG, and possibly make a proposal in the future.

Policies

It was decided to: 1) eliminate default complimentary registration for speakers, with the program chair being allowed to comp a registration on a by-request basis for events with anticipated attendance of over 300; and 2) increase the office's petty cash fund account limit to \$6,000.

Revised STG/SAGE Resolution

Nemeth said that paragraph 4 of the the resolution passed at the previous meeting need to be changed as follows:

4. That the USENIX Association will administer an election for the SAGE board of directors. Nominations for that board will close at noon on Thursday, October 22. All members of records on November 11 are eligible to vote. The new board will take office January 1, 1993. The board will consist of 7 members: the past president and 6 elected directors. In the first year, 3 members will be elected for 2-year terms (top 3 vote getters), and 3 members elected for 1-year terms (next 3 vote getters). Subsequent elections will turn over only 3 members of the board each year.

Other Business

Johnson felt we need to do something about NT. Grob said she has been actively trying to get someone to give a tutorial at the next Microkernels symposium. Grob took on an action item to increase NT awareness and involvement in USENIX. Grob also suggested that we consider offering spouse and family memberships, and Young would make a proposal.

BSD/USL

After discussion, it was decided to send the following letter regarding the USL/UCB lawsuit:



USENIX Association

UNIX System Laboratories
190 River Road
Summit, NJ 07901

Chancellor Chang-Lin Tien
200 California Hall
University of California at Berkeley
Berkeley, CA 94720

Dear Mr. Pieper and Chancellor Tien:

The USENIX Association's membership has been saddened and concerned by the USL/UCB lawsuit. We would like to offer any help we can to assist in resolving this matter in a way that removes the chilling and divisive influence this suit has had on our community.

Since the UNIX operating system has been widely distributed to universities, we believe it is essential that USL, universities, and students clearly agree on the extent to which exposure to the UNIX system in the classroom compromises students in their later professional work.

While the court process may eventually lead to such a resolution, we are concerned that the extended time scale will be chilling at a time when UNIX vendors must forge relationships to respond to new market forces. There is significant risk that the legal process will not fully explore the technical issues, and/or be influenced by legal technicalities that will not satisfy the legitimate needs of the parties.

Perhaps USL and UCB could enter into a technical mediation process to clarify and explore possible remedies. We would like to offer our assistance in any way that you would find helpful. For example, we could serve on, or help to organize, a committee of technical experts that might help both sides resolve this issue. The role model for cooperative resolution of differences between two of the premier organizations in the UNIX arena could only strengthen our industry at a critical time.

We stand ready to assist you, and would urge you to move towards a quick and equitable resolution of this matter.

Sincerely,

Stephen C. Johnson
President

cc: Marshall Kirk McKusick, CSRG
Dennis Ritchie, AT&T Bell Laboratories
David Hodges, Dean of CCEE, UCB
Joseph Cerny, Provost for Research, UCB

[See pg. 9 for response from USL - EY].



Roel Pieper
President and
Chief Executive Officer

UNIX System Laboratories, Inc.
190 River Road
Summit, NJ 07901-1400
908-522-6516
908-522-5171 (fax)
email usl@roel

December 19, 1992

Stephen C. Johnson
President
USENIX Association
2560 Ninth Street
Suite 215
Berkeley, California 94710

Dear Mr. Johnson:

Thank you for your offer of help. I would ask you to help us in correcting the misperceptions that currently exist in the UNIX® community concerning the issues you mentioned in your letter. However, before I address this, I want to respond to your concern regarding the litigation between UC Berkeley and USL. I assure you that USL did not enter into this litigation lightly. Quite the opposite, we sought to resolve our differences with UC Berkeley through several alternative approaches. Our efforts were continually rebuffed, ultimately leaving us no option but to file suit. Even now, we would prefer to arrive at a cooperative and amicable resolution that would allow us to avoid expending considerable resources and energy that could certainly be better applied elsewhere. Nonetheless, we are obligated to protect our business assets, the interests of our customers and the integrity of our technologies.

Clearly, there is a misperception concerning the nature of the lawsuit and the implications of exposure to the UNIX system source code. The fact is that USL is seeking to enforce its license in exactly the way that has always been applicable. Someone who has been exposed to USL's confidential information has always been required to keep that information in confidence. But that does not mean USL would try to stop people who have seen USL source code from working for a competitor or independently developing competitive software. It merely means that in that job they will not be permitted to use USL's confidential information or to create code that is copied or derived from UNIX System source code. And, the issue will normally arise only when, in that job, they produce software substantially similar to UNIX system software.

As you well know, the history of the UNIX operating system is one of cooperation and the collaborative efforts of individuals and teams throughout academia and industry. This is what makes the UNIX system unique. I encourage your support in helping to correct any misperceptions, so that we do not lose the most valuable asset that we share in this industry- the trust and confidence of those who apply the technology, create the applications, and deliver the quality solutions and products that are at the very heart of our business.

Regards,

A handwritten signature in dark ink, appearing to read "R Pieper", written in a cursive style.

Roel Pieper
President and CEO

cc: Chancellor Chang-Lin Tien



SAGE Election Results

The results of the elections for the Board of Directors of SAGE, (the USENIX Association's Special Technical Group – the System Administrator's Guild) for the 1993 & 1994 term are as follows:

Directors: Elected for 1993 & 1994, two year term

Steve Simmons	118
Pat Parseghian	115
Peg Schafer	111

Directors: Elected for 1993, one year term

Pat Wilson	107
Carol Kubicki	103
Paul Moriarty	99

Appointed Director: Elizabeth Zwicky - Past President

Not Elected:

For Director:

Paul Evans	88
Steve Romig	73
Pete Cottrell	70
Arch Mott	70
Lee Damon	61
Hal Pomeranz	50

Total number of ballots cast: 196

The new board took office January 1, 1993. The board consists of 7 members: the past president and 6 elected directors (above). In this first election, 6 directors were elected. In this first year, 3 members were elected for 2-year terms (top 3 vote getters), and 3 members elected for 1-year terms (next 3 vote getters). In deference to a strong desire by the USENIX Board of Directors for continuity, it was agreed that Elizabeth Zwicky, the interim president appointed in 1992, will automatically serve her remaining one-year term on the first elected board. In subsequent elections, alternately three and four members will be chosen for two-year terms.

The SAGE Board of Directors will choose its own officers after each general elections (every year).

System Administration Tools

System Administration Tools Your Vendor Never Told You About: The Chocolate Chip Cookie

by Elizabeth Zwicky
SRI International
<zwicky@erg.sri.com>

Our system administration group stocks the user services desk at irregular intervals with chocolate cookies, doughnuts, or other goodies. Primarily, this is a reward for the system administrators (in fact, we have convinced many of the users that bringing us chocolate chip cookies is a good way

of showing us we're appreciated), but it also serves to bring random people into the user services desk for a purpose other than complaining or asking for help.

Chocolate chip cookies are a tangible way to prove that somebody does care, and they are cheap enough to be provided whenever it seems necessary. There is also a lot to be said for having food lying around, since an administrator who is desperately trying to fix a machine is unlikely to stop for food, but low blood sugar rarely sharpens problem-solving skills.

System Administration Tools Your Vendor Never Told You About: The Logbook

by Steve Simmons, ITI
<scs@iti.org>

Once in a while I teach system admin, and number one on my list of things the admin must do is keep a logbook. Online is nice, but you can't take it with you on the plane, from client to client, into the bowels of the phone closets, etc. If it's not where you are, it's worthless.

Having the notebook on hand while you're working and are unable to get to a terminal is important in getting the data written down as well. My experience has been that if you don't write it down while you do it, you don't write it down.

Let's not mention the uselessness of on-line notes for diagnosing a down machine.

My logbooks go back seven years and are a great source of data. I tell my clients, "I'm not brilliant, I've just got a good memory." Anything which helps that memory is a plus.

Another good point to the logbooks is psychological. I've had several cases where a dispute arose over what was requested. The first time you pull out the lab notebook to settle the issue is usually the last time. This once got one of my bosses to start doing it to his boss.

Here are some good physical qualities of logbooks and why:

Hardcover. This book will get a lot of abuse, and be a valuable historical document. You need the strength. Reinforced corners and spines are a plus, as is a sewn binding. The binding must shed coffee or water (curiously, stationery stores are reluctant to let you test this).

Numbered pages. This is the easiest way to do indexing and make cross-references, for example, "See book 6, 23-29."

Not too many pages. The book has to be light enough not to be a burden. On the other hand, there should be enough to last a reasonable length of time. I used to aim for a year, but found that it required too large a book—300 pages. Remember, when you finish one notebook you'll be carrying both the old and the new around together for a while. One 300-page notebook is manageable, two are not. Your page count will

depend on how much you write. 150 pages lasts me about six months, but I'm not as terse as I was seven years ago.

Sturdy paper. It doesn't have to be acid-free or incredibly expensive, but it should resist the accidental tear. It should also give you a few seconds' grace for the occasional coffee or water spill.

Lined, not graphed. For about 4 years I used books with facing pages which alternated lined and quadrille. One day I went through and found about two drawings per book that actually used the quadrille. Everything else that the quadrille pages were used for was harder to read than the lined pages.

Left margins. This is where date, time, and location get noted, making scanning for dates and places real easy. Note both the start and finish time of activities. There's nothing more valuable than real data.

Size. I like 8x10. I've tried 8.5 x 11, 8.5 x 14, and various smaller sizes. It should be large enough to write about a topic or draw a picture easily, yet fit on a lap or small table when open. If it's too big it tends to get put on a shelf because it's in the way when open. If it's too small it's hard to write in, and a single topic is likely to get split across multiple pages. The 8x10 book is just large enough that you can cut the margins off a 8.5x11 photocopy and tape it into the book.

Right now I'm using the 150-page hardbound Boorum & Bease journals, model 21-150-R record ruled. They are well-made, take a lot of abuse, shed coffee and water, look good to a client or the boss, and fit nicely in a briefcase or on a desktop. They cost about \$30, and are worth every penny. If I could wish for one change, it'd be to make a recessed/embossed area on the front I could glue the business card into.

Prepping a new book is a simple process. I tape a business card to the outside front cover, the inside back cover, and the first page. On the inside back cover and the first page go a list of phone numbers where I might be reached and a request to call collect should the book be lost. The first sheet of the Boorum & Bease is an extra-sturdy sheet for table of contents. On this page I note the start date of the notebook, its sequence number, and leave a space for the end date. Below this is taped a small calendar (output from `cal`, actually.)

Getting into the logbook habit isn't particularly difficult. The hard parts are remembering to carry it around with you all the time and getting into the habit of noting things down. But once you've done it a while, it becomes second nature. After it's saved you a few times, you'll wonder how you ever lived without it.

Communication: An Important Aspect of UNIX System Administration

by Bjorn Satdeva

/sys/admin, inc.

<bjorn@sysadmin.com>

Have you ever stopped for a moment, and thought about what the job of the system administrator is about? Try to look past the backups done last night, and the new user account which was created this morning. What is the purpose behind the work which is done by the System Administrator every day?

The work of a responsible system administrator is to support the users on his or her systems, to assist the management in setting policies, and to help everybody to decide the future needs of the site. Often, a system administrator job description is a number of technical tasks, such as as doing backups or creating new accounts. I have even seen a description of the helpdisk function which was done as a list of technical tasks. In my opinion, this is not appropriate, as the tasks of a system administrator are only in part technical. The other part is to deal with other people, in the form of both users and managers. The fact is, the system administrator is often in the particular situation of needing to not only manage the users, but also management. Being in the position of managing both upwards and downwards, without being given much of the authority traditionally available to a manager, makes the system administrator's task load both unusual and interesting.

Adding to this is the fact that the system administrator is most often invisible to both users and management. As long as the computers and networks at the site are functioning well nobody will notice the ongoing work of ensuring status quo. However, as soon as problems occur, the sysadmin will often be the guest of honor in the "necktie party" which follows. In at least two cases I have witnessed system administrators losing their jobs because of problems which had a major negative effect on the user's ability to get work done. In each case, the main fault of the system administrator was to neglect to highlight existing problematic situations, brought about by attitudes and actions of users and management.

Therefore, it is of major importance to establish good rapport with both management and users, and to keep a good line of communication open with individuals of both groups.

One of the ways the UNIX system administrator can provide this is by always being ready to listen to and act on the problems in the user community, and by clearly communicating situations and requirements which may affect the operation of the system to the company management.

Several years ago Max Vasilatos reported that she had found her users to be significantly more satisfied if she spent some time every day walking the hallways, with the specific purpose of talking with them, rather than spending that time working on some of the problems needing solutions. It is therefore a good idea to make it part of the system administrator's job description to spend a least two hours every week talking with users not well known to them.

As it has been shown, it is an important part of a system administrator's daily job to ensure good communication. However, such communication consists of many other items than just walking the hallways, chatting with the users. It is equally important to ensure that policies and procedures are written down and published. Such documentation does not always need to be technical. Often, easily understandable and clear statements of the policies and procedures which have been established will get the message across.

In this article I will discuss some of the methods which can be used to establish the necessary rapport with users and management, both to provide the necessary written documentation, and to ensure good day to day communications. I will explore these methods first in the relationship to management, and then to the users. It is done in this sequence, not so much because management is a more important factor than the users, but rather because this is the sequence in which these relationships must be established. Any attempt to build strong relations with the user group can easily be undermined by a few negative individuals, especially if they are part of the middle management group.

The Context of Communication

Today, at most larger UNIX sites the user base is either engineering staff at commercial organizations, or students and faculty staff at educational sites, even though some data processing areas traditionally done by mainframes are starting to move towards UNIX. My experience as a UNIX system administration consultant has been mostly with larger commercial organizations. The experiences described in this article are therefore primarily applicable to such sites. However, I would expect much of the information could also be applied towards other kinds of UNIX installations.

At times, when I start working with a new client, the UNIX systems are in a state of chaos, sometimes with just a single system administrator trying to keep the systems working. At times it can hardly be characterized as proper "administration;" constant fire-fighting seems to be a much better description. Such a situation is sometimes made worse because the manager has minimal understanding of UNIX system administration, but still interferes in the day to day work. This can be further aggravated, if the managers background is in MIS. A person with this kind of background will often see UNIX as a maverick system which is hopeless to administer. Unfortunately this is not totally without justification, as some problems I have today, dealing with UNIX, did not exist in the old days when I was working with mainframes (although, to me the mainframe operating systems I have worked with seem mostly braindamaged). The biggest challenge in such situations is not the technical problems, but rather the attitude of many of the users, who often have become convinced that the UNIX support staff is the last place in the world they can expect to find any real help with their system problems.

It is therefore important to realize that to succeed in cleaning up the UNIX system problems, full support of management is necessary, both because their help is required to communicate to the users that an improvement is underway, and when necessary, to put down the foot to enforce unpopular decisions.

If the manager has limited understanding of UNIX system administration, it must be part of the system administrator's responsibilities to inform and educate middle and upper management about what needs to be done. One of the ways which has worked for me in the past is to write a number of memos, outlining the strengths and weaknesses of UNIX in a number of areas

such as backup and security, followed by recommended actions. However, for this strategy to work successfully, it is imperative that the memos be distributed to all interested parties.

When a good working relationship with management has been established, it is time to start to work on an understanding and agreement on the requirements for the site. When this is done, it is time to begin writing documents which, in an official manner, describe how the UNIX site shall be run. Most often, one of the first documents I work to get an agreement on is a Site Policy. Such a document describes, in general and non-technical terms, the policies which both the system administrator and the users must follow when using the site's equipment and software. Because it is very general, it is also short, and can therefore be completed in a short time.

When the Site Policy is done, the necessary framework for building reasonable policies for backup, security, disaster recovery, e-mail, USENET/Internet access, or any other policy deemed necessary, can begin. This process is often slow, as it is important to get acceptance from both user and management of such policies. However, as they start to emerge, many problems can be put to rest.

Again, it is extremely important that the policies be made generally known. The best strategy seems to be to have them online at a well known location. Bud Howell posted some small shell scripts to USENET about one year ago, which are helpful in making policies be reasonably well known. These shell scripts provide a simple policy reader, and also ensure that any new user will be presented an overview of current policies, during the first login.

Status Reports

I want to encourage every UNIX system administrator to start writing weekly status reports. Such a report will serve many different purposes. First it is a formal method to inform the manager about progress with ongoing projects, what tasks have been completed, and what is needed for the system in the near future. They can also be used to convey any outstanding problems which require a solution beyond the scope of the system administrator's responsibility. Most important, any outstanding problems must be stated at the end of the report *each* week, until resolved. Such problems can, for example, state the need for additional help or equipment.

This approach has been a life saver for me at several times. One such example comes from when I

was working for a small startup company. There was not a sufficient number of tape drives on the UNIX system, and it was therefore impossible to provide adequate backup coverage for the systems. I listed this as a special and important problem on my weekly status reports to my manager. However, the top management of the company was reluctant to spend the large sum of money which was necessary to implement a functional solution.

When a disk drive was lost a couple of months later, there were no recent backups of that drive, and the company suffered an estimated loss of over 800 hours of engineering time. This was aggravated by the fact that the disk crash happened two days before a major release. Unless I had consistently turned in my weekly status reports for the last two months stating the existence of that specific situation, and continued to request funding for additional tape drives, I would probably have been directly blamed for the loss of data (some people call this kind of status reporting method "CYA").

Communication with the User Community

While it is important to keep a good rapport with management, it is equally important to keep a good relationship with the user community. The work with management and with the user community are not two unrelated processes. Even though I have described them separately, they are very closely interconnected. Many of the decisions made by management and the UNIX system administrator must be based on input from the users.

The important factor here is "the users". They are often seen as a gray mass of disturbance. A system administrator who is able to remember that the users are individuals, each in many ways dependent on the computer in their daily work, will be able to create a better working environment for everybody. I have seen, more than once, a UNIX system administrator who had ruined the reliability and security of the site by being unresponsive to the users. When a system administrator is not supporting the users, then they will attempt to bypass him/her in order to get their work done. If the system administrator has created an environment where the security of the system is under constant attack from users, because they otherwise cannot get their daily work done, then it is the system administrator who is responsible for the security violations, although indirectly.

By keeping the users informed about the current state of the systems, and being doubly sure to communicate possible problems and scheduled downtime, it is possible to create an environment where the users are more understanding of the occasional need for downtime or reorganization of the systems. In fact, it has been my experience that most users are reasonable people, who often will accept even severe restrictions during change of the system, if they have been notified well ahead of time, and it has been explained how and why the new solution is for the good of the user community at large.

Of course, it is not possible to fulfill every user's request. Whenever I find I have been presented with a request which cannot be fulfilled, I make sure to have a meeting with the involved users to explain why the request had to be rejected. At the same meeting I also present what I think could be a workable alternative (if such a thing exists). In most cases one or more of those alternatives will be acceptable to the users. In cases where no alternative can be found, I always finish the meeting with a suggestion that the affected users should contact their direct manager in order to have the problem escalated in the management hierarchy.

By doing so, I hope I have ensured that the users know they have been heard, and that they have an avenue to continue pursuing their request. By notifying my own manager, I have given him/her a chance to be able to support (or reject) my position. In many such cases, the item of discussion involved purchase of more equipment, and the whole discussion had in this manner been escalated to a higher level of management, where such discussion belongs. In the cases where the users are unreasonable, the escalation of the problem should hopefully defuse the situation. However, this can only be expected to work well when good policies are in place.

Day to Day Communication

UNIX provides a number of tools which help the system administrator communicate with the user community. Most of these are well known and used at most sites. The classical method is of course e-mail, and if at all possible, this should be the official method of communicating events, like for example system down time. Unfortunately, this is not always possible. I have been involved with system administration at several sites, where most users did not read any e-mail, or where many non-UNIX systems were used, and no e-mail gateway existed between the various systems. One alternative to electronic mail is to

use broadcast messages for phone voice mail, if the organization has the necessary equipment for this.

Another classical method is to enter messages into the Message Of The Day file (*/etc/motd*). While most users today are using X windows, and therefore are seldom logging into their workstation or X terminal, this is still a good place to document system changes. However, as a practical manner to communicate upcoming events to the user community, the */etc/motd* file is becoming almost useless.

A third method for communicating this kind of information is found in UNIX System V with the news command. This command allows the system administrator to maintain a number of files with information about system status which is shown to the user by executing the news command (this news command should not be confused with the Usenet news). This command can easily be reimplemented on systems which don't have it, however, it is only useful if the users are trained to get information in this manner (it was originally intended to be called during login, but this strategy has the same problems as the use of the */etc/motd* file).

Therefore, besides using these methods for communicating with the users, it is a very good idea to start a small newsletter, which can explain upcoming changes or existing policies. Most important, the users must approve of upcoming changes. While it is a much slower process of making changes this way, it is also the way where good communication and cooperation can be reached. This does not imply that every user, however incompetent or unreasonably demanding, needs to agree. However, such agreement should be sought from key users. In my definition, key users are the people who are competent and well respected in the local user community. If the system administrator is able to establish good rapport with this group of people, many problems are already halfway solved.

In the real world, there are both incompetent managers and unreasonable users, but a competent system administrator can still create a reasonably well functioning site in spite of such people. It will require, however, that much emphasis is placed on adequate communications.

SAGE Book Reviews

Help! The Art of Computer Technical Support,
Ralph Wilson (Peachpit Press, 1991, ISBN
0-938151-14-2) \$19.95, paperback

Reviewed by Tonya Engst
TidBITS Editor
<info@tidbits.com>

I've earned a living through supporting or selling computer software and hardware in one capacity or another for almost five years now, and have always been bothered by the paucity of materials about the field. Some professions have large libraries devoted to them, but I've never run across a So-And-So Memorial Library of Technical Support. This may be so because support folks are so overworked that we never have time to write about what we do. At any rate, I eagerly awaited the arrival of Peachpit Press's *Help! The Art of Computer Technical Support*, and read it from cover to cover in short stints over a period of two days. One of the problems with being a tech support person is that you may end up with your productive time broken up into about fifty two-minute blocks over the course of a day, which does weird things to your personality after a while.

Help! could be yet another pop-business book about using cute psychological tricks on your customers ("Don't sit across from them at a table; sit next to them or sideways from them to make you seem friendlier.") or it could offer tired, simpering maxims such as "The customer is always right" and "Never make excuses." I learned these at a support seminar, but I promptly discounted them because I know darn well that the customer often doesn't have a clue ("I don't need a PostScript printer; I only print from PageMaker."). As for not making excuses, just try working for an educational reseller of Macintoshes and not make excuses when Daddy calls from Long Island to find out why his daughter cannot purchase a computer until she actually registers for college. This book assumes that the reader has a brain and has mastered the basics of spitting out the chewing gum before answering the phone.

Help! will assist everyone involved in computer support from high-level managers to the most overworked techs in the cubicle trenches. It's for people involved with consulting firms and internal help desks, as well as software and hardware

companies that support what they sell. Wilson offers ideas and examples about improving support on all levels, with plenty of real life examples and quotes from leaders in the support profession.

For suit-types, the book discusses what personality traits make for a good support person, how to train support personnel, how to keep techs from burning out, and how to cost-justify your existence. For those managing phone support centers, it discusses various ways of charging (or not charging) customers for support. You'll find out WordPerfect's rationale for providing toll free support, why Ashton-Tate provided some support for the cost of a phone call, and the argument for and against 900 numbers as the emerging phone support method. Help Desk managers may be interested in the discussion of the pros and cons of "outsourcing," or making someone outside the company do some of the work. One chapter analyzes and explains the main features of several commercial databases used to store technical information and track customer information.

People who actually talk to customers and provide support will find useful suggestions for most aspects of their jobs, from assisting difficult customers to graciously accepting feedback. Wilson has done his homework here, with suggestions for dealing with all sorts of customer situations including skeptics, four-letter abuse, and "Novice Users and the Terminally Confused." A particularly valuable chapter is the one on the development of trouble-shooting skills. Wilson discusses the difference between internal and external support and even looks at alternative methods of support such as fax and email. I had the most fun with the last chapter, though, which discussed how to behave as the recipient of technical support. If only my callers would read this chapter before calling me!

Ralph helpfully included a bibliography of related materials, which I hope to look up in the future. After reading the book, I had some new ideas for working with users and a better understanding of the different aspects of providing support. Peachpit's books tend to be fun and informative, and *Help!* lived up to my expectations. Unlike some other Peachpit books that feature extreme brevity, this book is a solid 200-plus pages, and is worth the \$19.95 sticker price. Highly recommended.

[From TidBITS#145/05-Oct-92. This article reprinted with permission from TidBITS, copyright Adam & Tonya Engst. TidBITS is a free, weekly electronic newsletter that covers the computer industry with an emphasis on the Macintosh and on electronic communications. For more information, send email to <info@tidbits.com>.]

UNIX System Performance Tuning, Mike Loukides (O'Reilly & Associates, Inc., November 1990, ISBN 0-937175-60-9) 313 pps, Paperback, \$24.95

by Steve Simmons
Inland Sea
<scs@lokkur.dexter.mi.us>

UNIX System Performance Tuning is one of the first O'Reilly books intended for the system administrator. When told it was in process, I was very skeptical about the ability of anyone to do an adequate general treatment of such a complex topic. I was pleasantly surprised when Loukides proved me wrong.

Loukides presents the topic in simple and pragmatic terms – what is meant by system performance; what are the elements which contribute to it; and what kind of tools are available to measure the various performance areas.

He starts with the simplest and broadest measures of system performance – simple tools like ps, uptime, and so forth. Once these are established, he moves into a series of topics focusing on each of the areas of system performance. At each step he shows a variety of tools and indicates clearly which tools are available on which UNIX variants.

The book covers a fairly broad set of UNIX variants, primarily looking at four: SunOS 4.X, BSD 4.3, System V.2, and System V.3. As of his original publication date production versions of System V.4 had not been shipped, and Solaris had not been announced. This lack should not deter anyone from getting the book; in any given area at least one tool from BSD, SunOS, or System V.3 should be available under both System V.4 and Solaris. In many cases Loukides had some preliminary data about System V.4 and makes suggestions about specific points to address.

The advice offered is usually timely and to the point. Readers are cautioned to read an entire chapter before attempting to follow through on tuning any one area; sometimes the best advice is found in the summaries which close each chapter.

Those looking for simple answers will not find them in this book. System tuning does not admit simple answers, and Loukides is careful to note this. There are few magic bullets that will make machines run faster, and those few apply only to some well-defined easily recognizable situations – memory starvation, spindle overload, etc. Once these have been checked and fixed, Loukides stresses that any system tuning is an ongoing cycle of monitor, analyze, adjust, and repeat.

Loukides repeatedly points out that improving system performance is not simply a matter of making the machines do their existing jobs faster. Sometimes the best way of improving overall performance is by change job mix, job schedules, or user activities. He offers a number of areas where one can detect system activities which degrade performance and discusses changes which could be made to improve the situation.

The experienced system manager who is relatively new to UNIX will find this book invaluable. People with good groundings in operating systems will find the book equally useful. Simply having the data on tuning and analysis tools gathered together provides an important service to both groups. These people will find the descriptions of each performance area useful primarily as they relate to UNIX, not as discussions in themselves.

Two areas where Loukides is somewhat deficient are the difference between disk read and disk write performance, and NFS. O'Reilly regularly updates their material; one would expect this will be dealt with in future editions.

In summary, this is a useful book even for the most experienced system administrator. I learned of several tools I had not encountered, and when the day comes that I'm exposed to some of the other UNIX variants I will have a ready source to identify the appropriate tools. The less-experienced manager will not fare as well, but will still be able to find coarser bottle-necks and apply the simpler fixes. With sufficient work and ability to experiment, they will eventually be able to do more subtle tuning.

SAGE: A Proposed New Model

A Proposed New Model of Large Site System Administration

by Peg Schafer

BBN

<peg@bbn.com>

[SAGE editor's note: This is a condensed version of "Is Centralized System Administration the Answer?" LISA '92.]

Today's standard model of centralized system administration originated in the ancient days of THE big mainframe in THE machine room. The theory and practice of centralized control has been the foundation of policy development, user services, hardware configurations, network management, and software development. Indeed, the concept of centralized administration is so basic, it is the default mode of thought in discussions of system administration.

The Yin and Yang of Centralized Control

Historically, there have been many advantages to centralized control. Users were able to work with one closely knit group as it provided one point of contact for requests, complaints, and questions. As sites grew, and as machines became more numerous, new policies were built as extensions of the established single mainframe policy.

As support services enlarged to meet the needs of the new distributed technologies, specific areas of expertise developed. A centralized support group will now consist of experts in separate areas of system administration, e.g., network, news, mail, nameservers, Appletalk, etc. Indeed, many centralized support groups have internal development groups which do not support users, but create the necessary toolset required for centralized support. But let us not forget the most important reason for centralized support: it is cost effective.

Despite the advantages stated above, centralized control may not be possible or the best solution. Cookie cutter techniques make deviations from the standard system configuration very difficult; non-standard systems are left free-floating – lost at sea.

The model of central control may not match up with the corporate structure of a company. Due to internal conflicts over goals, resources, and per-

sonnel, some divisions may resemble "The Warring States of Greece" rather than one company united in a common pursuit.

Large system administration groups have a tendency to develop baroque methods of work, which result in inordinately long response times. Specific questions on local applications may be routed through a number of people before the answer is determined.

Diversity is Now the Name of the Game

Today, centralized support groups are required to maintain an ever expanding array of systems for every effort in the company; e.g., research, software development, manufacturing, MIS, document production, etc. Centralized control may not be able to meet the specific needs of these systems. Cookie-cutter techniques and cloning methods may not be adaptable to the variety of hardware platforms and unique configurations. Research will have the newest of the new machines with a high turnover rate. Depending on the area of research, machines will have special purpose (and often VERY buggy) hardware and software. Software developers are driven by release schedules, hence are sensitive to the timing of any system changes. Their machines range from beta software and hardware platforms to the oldest of the old. They will have at least one of everything -- a very heterogeneous environment. MIS departments are completely different animals altogether. Their management structure may view computers as black boxes which they utilize to get their very important work accomplished. Security is always a concern on these machines; not only from outside attacks, but strong measures must be followed to prevent intrusion from within the institution.

Ownership of machines within an organization can contribute to the diversity faced by centralized system administrators. The support task is greatly simplified by a homogeneous collection of hardware platforms. However, different groups or divisions in an organization are most likely to purchase equipment tailored to their own needs. Funding is always a determinant. One group will have money to buy new machines, while others may have to cut back. Within such organizations, there is little hope of maintaining a standard system or hardware con-

figuration. In addition to this diversity, corporate structures can be barriers to centralized control and support. Management's conscious decision not to allow non-local control is a valid alternative.

In sum, as centralized system administrators are required to support an ever increasing variety of groups, applications, and machines, their divergent (sometimes conflicting) needs are pulling the central services in multiple directions simultaneously, making the job nearly impossible.

The Proposed Model

I believe the future of system administration does not lie solely in the development of central services. Rather it lies in the co-operation of central services with local system administrators who, in turn, provide the primary support for their user groups. I propose a system by which administration responsibilities are shared between a central group and a local system administrator for each group. I believe there is a role for centralized administration; I also firmly believe there are a range of services which can only be supplied efficiently by a local system administrator.

Let us start with some basic definitions. "The computing community" is an interacting population of individuals of all skill levels, or a group linked by a common policy, in our case, a large company. "The computing environment" refers to the machines and the hardware and software configurations used by the computing community. "A Local System Administrator (LSA)" is a person who works within a group, and who is completely familiar with the computational requirements of the group. "Central services (CS)" provides support for the total corporate computing environment, and in particular, can address the needs of the Local System Administrator.

The Proposed Role of Central Services

Central Services organizes the diverse groups and supplies them with information necessary for their continued development. The CS must direct its efforts toward the the development and enhancement of the computing environment.

Policy and standards generation is a key objective. CS is responsible for development of the policies in conjunction with the LSAs and the computing community. Timely revision, enforcement, and arbitration responsibilities fall to the CS. Policies and standards can cover every facet of system administration from policies on security to standardized rc files. Standards in areas

such as file system layout and naming conventions can be agreed upon and distributed by the CS. Local system administrators may deviate from these standards as required to fulfill the computing requirements of their group.

The network is the lifeline of computer environments. Network administration, both hardware and software, belongs in CS. This group would maintain the gateways to the world, and critical network boxes. All global network services fit in here – Domain Name Server, NTP Servers, etc.

Large databases accessible by the whole company are maintained by the CS: NIS global administration, company phone book, information services, dictionary servers, finger databases, libraries, license servers, etc.

E-mail is a service to the total computing community, hence anything connected with it is under the domain of the CS: mail machines, sendmail.cf, /usr/lib/aliases, etc.

Distributed printing services are a big headache everyone is willing to give to anyone else who will do it. As one manager put it, "If you were evil in a past life, in this incarnation they put you in charge of the printers." In the model presented here, the CS would be responsible for the configuration and maintenance of the printer hardware and software as it is another global computing service.

Security is of great concern for all. The CS group can monitor the network, spot-check machines, advise the community on new security features, coordinate security alerts, evaluate new operating systems for security configurations, advise LSAs on the proper security procedures, provide a site representative to CERT, etc.

Information gathering and dispersal to the user community is critical. CS may publish a computing guide for users. This guide would contain pointers for more information, advise on the resources available, explain policies, etc. But the CS should also provide forums for discussion of critical issues by the user community.

The organization of vendor presentations and evaluations is very helpful. Not only can the CS examine and evaluate new hardware and operating systems, they may host these evaluations and advocate evaluation by members of the computing community. Indeed, it is likely there are members of the user community with relevant expertise; their evaluations and recommendations can be solicited and distributed.

Since computing technologies are turning over at an ever-increasing rate, continuing education of the computing community is essential. The CS can coordinate classes, tutorials, and presentations on many topics including new methods of computing, new system administration techniques, etc.

Members of the CS may serve as official representatives of the company's interests in external organizations having to do with industry standards and user groups (e.g., POSIX).

Backup and tape archival services are classic CS services. New centralized backup systems (e.g., Legato) are amazingly efficient and off-load this tedious task from the LSA.

Site licenses and contracts for software and hardware can be handled efficiently by the CS. Negotiating for a discount based on volume is always welcome.

Some groups request help in the evaluation of their computational requirements and selection of hardware for purchase. This complements nicely the vendor presentations and evaluations cited above.

CS Support for the Local System Administrator

In addition to the functions listed above, CS communication with and support of the LSA is vitally important. The CS can develop software tools for the LSA such as automated installs, site specific accounting programs, new user scripts, etc. The CS can provide backup support for LSAs. A visiting system administrator can be on hand while the local system administrator is out for conferences, training, vacation, etc. The LSA can rely on CS for help with a particularly perplexing problem; sometimes two heads are better than one. When 25 new machines just roll in the door, "high tide" services are always welcome.

The Role Of The Local System Administrator

An emerging role of system administration is now in the management of local application software. Often the LSA is the tool maintainer. As mentioned above, machines and the applications software have diverged to such an extent that detailed knowledge of the utilization of the machines is a key factor in day-to-day maintenance and problem resolution. The LSA must have a detailed understanding of the layout of their machines and of the specific applications resident on their machines.

It is important that each group recognize the need for a LSA and appoint a person who has an ade-

quate skill set to fulfill the requirements of the job. There is often a wildly divergent range of knowledge of computers between groups. The role of each LSA depends upon the computational needs of their respective group.

For example, a group whose research interest is some facet of computer research (graphics, file systems, AI, etc.), and which possesses constantly evolving machines, will require a level of system administration which is extremely expert, as the user group is expert. If the group is large and has extensive administrative needs, the LSA may actually be a group of full time system administrators.

On the other hand, a group whose interest does not require modified systems, such as a documentation group, will generally utilize stable turnkey systems and will be populated by a user community with basic system skills. The LSA, in this instance, will be called upon to know more about the application software than the computer systems. Indeed, computer system support may require as little as 5% of the LSA's time, while local application support may consume a larger amount of time.

The point is that both groups require a unique level of service based on the type of work within the group, and the needs of both groups require adequate representation within the computing community.

In general, LSAs are the first line of user support. The LSA will be on site to answer users' questions in real time. The LSA can spot developing problems and respond immediately.

The LSA should also serve as the liaison between a group and Central Services. The LSA's representation of the local group's interests in all areas such as policy development, dispute arbitration, and allocation of resources, is essential. The LSA is responsible for informing the group of current trends and changes to the computing environment.

Finally, the LSA may be an employee of the local group or an employee of central services assigned to the group. Many groups prefer to have total control (i.e., hire and fire rights). Details can and do vary from company to company.

Cooperation and Communication are the Key

What is there to prevent confusion, disorder, replication of work, and a great bloody mess? Cooperation and communication between the LSA and CS.

First, there must be recognition of one goal: both CS and the LSA exist to provide the best system administrative support services. The feeling of unity and mutual respect must be fostered and supported. To that end, every opportunity for communication must be utilized. Mailing lists are essential. Monthly meetings with the user community are very helpful. Weekly meetings with the system administrators are necessary. Bboards can announce information to users, and when archived away, can provide a helpful record of events. Listen to good ideas – they can come from anywhere. Co-development of initiatives invites unity and understanding between groups. Initiatives on policies, security, and standards for the company may lead to unexpected benefits for the whole company. Organize working groups to come to agreement on company wide standards. Promote the sharing of binaries and other resources.

In a nutshell, I have proposed a system in which there is recognition of the job designation Local System Administrator for each group of machines. The LSA is responsible for the efficient adaptation of the machines to the computational task of the group. The Central Services group provides support for those services which are in common use across the computing environment, and also provides information and support services to the LSA. While I am a UNIX system administrator working at a predominantly UNIX-based site, this model is not restricted to UNIX-based computing environments.

Results at BBN

At BBN I soon realized standard methods of centralized UNIX system administration could never work due to administrative boundaries. The books and tutorials on UNIX system administration were based on the premise of centralized control of systems, and did not touch on the topic of distributed system administration responsibilities. Indeed, many of the papers which have been presented at past LISA conferences are on tools which enable centralized control over a variety of architectures, and/or a large number of machines. Hence, the development of this model.

The group of which I am a member, Distributed Systems and Services (DSS), is chartered to provide UNIX support on a contractual basis to groups within BBN. The divisions own their machines and exercise primary control over them. The DSS supports roughly 60% of all UNIX based systems at BBN. In addition, the DSS provides e-mail and printing support to the BBN computing environment.

Some groups which are not supported by the DSS have their own full time system administrative staff, while others have a part time system administrator, and still other groups do not have any organized support at all.

The DSS had discovered that quality support was increasingly difficult to provide to the larger groups, basically due to geographic separations, the requirement for immediate responsiveness, unique configurations, and the expanding demands of an expert user group.

In the LSA model cited above, the LSA could be a member of the local group, or a person from the CS stationed at the local site. In fact, the DSS stationed two representatives as LSAs in a software development group on a trial basis about a year ago. The group had extensive system administrative requirements which necessitated at least two full time system support staff on site. By all accounts the experiment has worked out very well.

To date, five members of DSS have local assignments. The DSS has a unique assignment plan to facilitate communication and coordination: DSS's LSAs are scheduled to work 4 days a week at the local site. The remaining day is spent with the DSS working on projects. The members of DSS who are assigned remote posts are also required to attend the weekly meeting of the DSS. With this representation, all DSS discussions include input from LSAs.

The DSS as the central services group is making efforts to promote system administration company wide. To facilitate contact between all system administrators and users, the BBN UNIX User Group meetings were initiated. Here, topics of interest to the BBN computing community are addressed along with announcements of changes of service, bug reports, information distribution, etc. A bboard has been dedicated for announcements from DSS and comment by the user community. The user community is encouraged to contribute to enhancement of the BBN computing environment via bbn-public. (See "bbn-public – Contributions from the User Community", P. Schafer, Proc. LISA '92). Company standards have been developed by representatives from all areas of BBN. For example, the user community and LSAs have called for a BBN-wide computer security policy, which is currently under development with representatives from each division. The DSS is constantly identifying services which are of interest to the total BBN computing community, and striving to fulfill these services.

Conclusion

In the course of writing this paper, I found myself using the terms "control" and "support" almost interchangeably. This is the crux of the problem – how can an organization support a machine if it does not control it? In general, groups want their own machines configured and supported to fit their needs, and are not concerned with the management concerns of other groups. While centralized control and support may work and continue to work well at some sites, it is not the answer for all sites.

I have proposed a model of system administration where the establishment of the position of the LSAs is essential. I advocate a strong Local System Administrator who is ultimately respon-

sible for the efficient operation of a group's machines. Communication between Central Services and the Local System Administrator is fundamental to quality system support.

While computing technology has become increasingly distributed, UNIX system administration has lagged behind. Hopefully this model will provide some insight into this problem and will facilitate the search for solutions.

Peg Schafer is the Senior Systems Programmer for the Distributed Systems and Services group at Bolt Beranek and Newman Inc., peg@bbn.com. The opinions expressed here are Peg Schafer's, not BBN's. If you wish to ask questions please feel free to send e-mail.

SAGE: Call for Tools Review

Tools for System Administrators: Reviews for SAGE newsletter

by Christine Quinn
Stanford University
SAGE Assistant Editor
[<quinn@ee-cf.stanford.edu>](mailto:quinn@ee-cf.stanford.edu)

How would you like to be on the SAGE newsletter reviewer list? Interested in telling others what you think of the public domain software that's out there? Interested in seeing your name in print? How about helping out a fellow sysadmin by installing and trying out one of the myriad tools available and writing about it?

SAGE is starting a new, regularly published column to review tools. Its purpose is to review tools for the system administrator in a way that assists the new and experienced administrator in deciding whether a particular tool is useful in a given environment.

These columns will include a general description of what the tool does as well as the reviewer's opinion on how useful the tool is and whether or not it's worth the install. It will also include the nitty-gritty details like where to get it and how well documented it is.

Tools of interest will at first be limited to non-commercial products. Since vendors do such a good job of reviewing their own and other's products (yeah, I know, not always fairly), and since commercial magazines like *SysAdmin* and *UNIX Review* make a point of reviewing the big name products, we're striving to fill the niche for these smaller, but just as useful, tools.

Examples of such tools may include, but are by no means limited to, security checkers, new user adds, mail packages, problem-reporting programs, software distribution tools, etc.

Why would you want to contribute a tools review article for SAGE? Think of the fame and fortune! Well, at least think of the fame! Seriously, you'll be helping your fellow admins and getting your name in print to boot. What more could you ask?

If you're interested in helping review tools, send email to me, Christine Quinn, [<quinn@eecf.stanford.edu>](mailto:quinn@eecf.stanford.edu) or Bryan McDonald (Chair, SAGE Publications), [<bigmac@erg.sri.com>](mailto:bigmac@erg.sri.com). If you have tools you would like to see reviewed (especially if you'd like to review them), send those in, too.

UNIX is Dead; Long Live UNIX

by Mike DeVaney

<dm_devaney@pnl.gov>

Whither the creative force in the development and extension of the UNIX OS? Commercial interests have ensnared this poor beast. Academics, hackers, and innovators should kill it – put it out of its misery – leave its carcass to feed the drones who slave for gold crafting SVRn, AIX, HP/UX, OSF/n, PowerOpen, A/UX, IRIX, and on and on, ad nauseum.

I thought the question should be where will be the next intellectual action in operating systems? Two inadequate possibilities came to mind.

- *micro-kernel UNIX variants* - just one step ahead of the sheriffs and marshals of commercial interests, soon to be captured and forever imprisoned, doomed to an endless future of mindless service without replenishment, renewal, growth, or viable progeny
- *pen-based OS* - we shouldn't confuse a fertile field for application developers with a fertile field for OS developers, which this is not

I dismissed these possibilities quickly and began to explore the right brain passages for some clues. I tunneled into a disturbing knot of busy neurons and tapped the flow. After some time I was able to assemble some random packets into what may be the proper sequence of a message, at least it seemed to me to be coherent.

The massively parallel personal daemon system (mpps) will perform tens of thousands of concurrent processes endlessly. These processes operate in the background, only occasionally coming to the fore to provide priority knowledge and information, or in response to random user acts. The necessary CPU power is provided by a few hundred to a few thousand commodity chips. Early models of a usable machine can be assembled cheaply today with a few hundred low-end 80386 or 68030 chips. Think how soon commodity chips will include Intel's P6 and P7, IBM/Apple/Motorola's PowerRISC-2, Sparc's Viking-2, DEC's Beta. RISC chips that operate at hundreds to thousands of megaHertz and individually offer staggering measures of processing power. So the answer is, yes, we could run tens of thousands of daemon processes.

What might these processes be, and how might they be managed? Shades of AI (gulp). Some of these processes should be operating as knowbots. Some of these processes should be operating as infobots. Some of these processes should be operating as network explorers. Some of these processes should be operating as health watchers – constantly guarding against infections and intrusions, repairing and circumventing damaged parts of the system, and so forth. Overall the system should exhibit behavior that remembers what its user does, and *strengthens* processes that frequently serve the human thing, and *weakens* processes that rarely are called upon. The system is constantly extending itself by adding new processes to its bag of tricks. Occasionally, the system even adds a kludge process of the ancient type once known as *applications*, usually in response to a special request from the carbon-based unit.

This highly personalized system has a great deal of information readily available to its user, and responds with a myriad of activities to stimuli from its user. The choice of activities for a given stimulus is a combination of wise choices, tempered with learned strength weights for this particular user. The objective is to be prepared to offer the user ready access to the next information need by using background processing to hurry along all the neural connections from the current stimulus, gathering what seems appropriate. In this fashion, the answer to the user's next stimulus will likely be pre-gathered (pre-computed) or already well in process. Also, as in a human memory, the user will be offered a selection of visual, auditory, and perhaps other sensual cues to somehow connected information as the various stimulated daemons discover interesting knowledge nuggets. A simple example scenario might be:

- User's eyes are determined to excite when reading certain passages of a news article
- The family of daemons who detect this phenomenon instantiate several infobot daemons (same daemon, different data) to search for related knowledge (information, data, articles, facts, statistics, pictures, historical and cultural background)
- As related information is found, the user may be provided cues to their existence if the relative

strength of association is thought to be great enough

- If the user specifically requests additional information, pre-fetched results can be made instantaneously available
- Additional cues are provided as the relative importance of other knowledge nuggets is raised, and still more daemons are instantiated in response to this stimulus
- As the user pursues related information, the weights of the associated daemons are *strengthened*, while daemons whose work goes unappreciated have their weights *weakened*.

An mppds should come with a large number of daemons tightly integrated into the system and must have a richly defined pluggable interface for dynamically adding an endless number of several daemon types.

This ought to require an interesting OS. Some of the work in artificial neural computing systems would seem to apply. Also, distributed computing systems, multi-media systems, advanced user interfaces, human computer interaction, and other areas of computer science and related fields. Yet the large envisioned scope of an mppds leads me to infer that all of these are mere baby steps – while we desire to fly through the air from midcourt and perform a six and half revolution behind the back through the legs under the arm-pit blindfolded double slam dunk.

How to avoid information overload? This question alone ought to keep many mppds designer/developers occupied through the wee hours of more than a few mornings.

I could (should – to hone the idea) go on and on, but trust this small view into the concept will suffice for now.

Response to File System Workshop Report

by Noemi Paciorek

<noemi@chpc.org>

and Marc J. Teller

The Center For High Performance Computing

Dear Editor,

We would like to respond to the Drew Perkins' File Systems Workshop Report published in the September/October 1992 issue of this newsletter and address some of the inaccuracies presented therein. We would also like to clarify some of the ambiguities found in the report. This response also references pages of the paper, "An Object Oriented, File System Independent, Distributed File Server," published in the proceedings.

The CHPC distributed file server (CHPC DFS) is an integral component of the OSF/1 AD release from the Open Software Foundation Research Institute. The CHPC DFS is a local area distributed file server that provides a single name space, with location transparency, in a distributed environment (p. 45). Traditional distributed file servers, such as NFS and AFS, do not contain adequate provisions for supporting a single name space (pp. 50-51). The CHPC DFS was designed to meet the needs of multicomputers, wherein several nodes cooperate to provide a single system image, and, thus, a single name space. However, this server also efficiently manages files on single node systems and collections of workstations connected via a LAN (p. 45).

Contrary to the description presented in Perkin's report, the CHPC DFS offers an evolved VFS interface that provides distributed, as well as local, file service in a file system independent fashion (pp. 49-50). These distribution mechanisms are in contrast to those of other file servers (e.g., AFS, NFS, RFS) that perform their own distribution. However, the CHPC VFS is backwards compatible, and thus other distributed file servers may run along side the CHPC DFS. Further, adding new file systems underneath the CHPC VFS model does not require providing additional distribution mechanisms because distribution is supported in a file system independent manner.

Each file server communicates with other servers and clients via the Mach IPC message passing paradigm and the server utilizes Mach ports to represent file system independent objects (pp. 52-54). As a result of utilizing Mach IPC, file servers do not require any knowledge of the protocol used to communicate with clients and other servers. Further details of the implementation may be found in the paper.

We believe the CHPC DFS offers features not readily available in other VFS-based distributed file systems. Its file system independent distribution mechanisms allow multiple file system types to automatically offer distributed, as well as local, file service. It also transparently enforces a single name space within a distributed environment. The CHPC DFS is currently being used to provide file service for multicomputers and networked PCs.

DOS Digs in Its Heels

by Bill Gallmeister
<bog@lynx.com>

Time to write the quarterly USENIX POSIX snitch report. Boy, is it nice to be able to do this kind of work at home. I just upgraded my home machine. It's a wimpy little 16MHz 386sx, but with 8 megs of memory, a 387, and a couple of fast disks, it moves along no slower than, say, a VAX 11-750 with three users on it. I've taken to calling my machine "Lenny" – not a real big brain, but strong as all get-out.

The REAL win, though, is the MKS tools that I got recently. This is nice. I can use the Korn shell. I can use *grep*. Most importantly, I can use *vi* to write things. Ahh. It's almost like being on a UNIX box. Simulated UNIX Veneer at a Small Additional Charge.

I write Snitch reports like everyone writes device drivers. Find the last snitch I wrote. Copy it. Edit it. Ooh, this is nice. I can now "cd work/posix/usernix/snitch" instead of "CD WORK\POSIX\USENIX\SNITCH". I can "cp" instead of "COPY". I can, uh, *vi* instead of...what's this? A file full of garbage – delete characters, little snippets of ASCII that have nothing whatsoever to do with my snitch report. *vi* thinks it's editing a binary file or something, informs me that this is one big two-line file with 16 non-NULL characters in it, immediately makes some hidden change so I have to :quit! instead of just :quitting. Yow, what a harrowing experience. At least we know that this *vi* has the appropriate UNIX response to the situation.

I guess the DOS word processor I used to use produces something other than ASCII by default. Well, that makes sense, I guess. Got to put those formatting commands in somehow. And heck,

this way it's hard to switch word processors. Of course, all the icky DOS packages have selections to convert their competitors' formats. Not so with *vi*, at least as far as I know – it's not on the *vi* quick reference card. I suppose there's probably a way to do it in emacs (C-s convert-from-random-dos-word-processor-format-please ESC), but my little motherboard maxed out at 8 megs, so emacs seems to be out of the picture. Well, heck, I'm a man, I've got AWK and SED now. I can get around this!

On second thought, maybe the faster thing to do would be to bop back into the DOS word processor for a second, a teeny little instant, and convert the file to flat ASCII. There. And I only feel slightly unclean. *vi*...yesssss! Houston, we have editing. Repeat, we have editing, over... Let's see, we've got all the cursor keys I know about. We've even got the full-on presence of ex pattern matching. Yummo. Feels like home. Just :wq, we're out and lpr the thing to read over.

lpr: not found. Damn! Well, it probably makes sense that DOS would evolve into System V, not BSD – lp: not found. Hmm. ls /bin. ls: File or directory "/bin" is not found. Well, I guess we know that MKS rewrote ls, eh? Oh, yeah. ls c:/bin. Nothing there. Well hell. cat snitch > /dev/ no no no. cat snitch > lpt1: and out comes the snitch report. Except I have to do a manual form feed to get the rest of it out. Yuck! It looks disgusting! Where are my nice fonts? Okay, okay. I give up. Back into the word processor, read in the flat ASCII, add some nice fonts, print it out. Ah. Looks nice.

All right, so my little stupid home machine isn't UNIX yet. At least I've got *vi*.

An Update on UNIX-Related Standards Activities

by Stephen Walli

Report Editor

<stephe@usenix.org>

USENIX Standards Watchdog Committee

POSIX – Caving In Under Its Own Weight

“Standards are commercial and international politics dressed up as technical arguments.”

I think POSIX is caving in under its own weight. All of the hard nuts-and-bolts work effort of defining a technical programming specification is slowly being mired in the mud. POSIX exists to “... support application portability at the source-code level. It is intended to be used by both application developers and system implementors.”¹

It has been floundering for some time in a mess of its own making. I want to look at this mess, describing it and its historical context, and offer up a few possibilities for solutions. This article is long, but there is a lot of context that needs to be understood to see what’s happening to an otherwise useful standards effort. The article ends with a list of e-mail addresses to which you may wish to send any questions and concerns. In fact, I encourage it, and hope that you’ll be convinced by the end of the article.

The Problem

There are two sets of people doing work in the POSIX working groups. The first set sit in the individual working groups, distilling historical practice and experience into a technical specification “for application developers and systems implementors.”

The second set of people have typically been involved at the working group level for quite some time. They are often chairs of the groups or other officers. These people have begun to have coordination meetings and form steering committees outside the working group structure. All of the pieces of POSIX are related to one another, and there is a genuine need to coordinate between the different groups of heads-down-over-the-specification-technicians. The bureaucracy has grown because of need rather than

desire to hold extra meetings. Most of the people involved can think of more enjoyable ways to spend their time.

I wander in these steering committees, sub-committees, and the hallways of POSIX. It quickly became apparent to me that this is where the politics that drives POSIX is most on display. I was eventually around long enough to get involved in some of these committees. (Fool me.)

There has been a strange tension in these rooms for quite some time, coupled with a terrible confusion and sense of apathy. This is not noticeable in the working groups themselves. Heads down and oblivious to the politics of POSIX, the working groups are buried in the religious wars and politics of their own technical specification.

A couple of POSIX meetings back, it began. First in one steering committee, then another, and another. The group would hit a crisis point, and throw up its hands. Despite the fact that each room contained people with a long history and knowledge of POSIX, they would reach a point of apparent confusion as to how to coordinate with another steering committee or sub-committee. (The running joke is that we need a steering committee steering committee, but it really isn’t seriously contemplated.)

Finally, someone would suggest we need to define the problem. I offered to go away and write it up. (More fool me.) Then the next sub-committee meeting. The same process. Tension, confusion, “let’s define the problem.” It started in the Project Management Committee. I later saw it in the Steering Committee for Conformance Testing, then the System Interface Coordination Committee. These are all really fundamental sub-committees, with a lot of POSIX history in their membership.

The coordination complexity is amazing. The major areas of POSIX requiring coordination are the base documents themselves, their test methods, and their structure with respect to language independent specifications (LIS) and programming language bindings. (This complexity has spawned profiles, about which I’ve yelled enough for now.)

1. ISO/IEC IS 9945-1:1990,1.1 Scope, p.1, lines:2-3.

Steering committees were thought to be a way out of the mire. If we just communicate with one another, the problems will all become apparent, sort themselves out, and go away. But ultimately this falls down. POSIX is too big. The steering committees have no authority to impose their collective will. POSIX is a volunteer effort. There are no sticks and there are no carrots.

If it becomes too much trouble to build the standards, then the volunteers will cease to arrive at the meetings. The POSIX standards effort will fail. Or worse yet, they will continue to be defined by fewer and fewer people with sound technical background and a proper perspective on the subject. This will cast doubt on the good work which has already been done.

Test Method Madness

To ensure that implementations of the POSIX.1 standard could somehow be tested and certified in a uniform way, the POSIX.3 standard (Test Methods) was created. This work was heavily supported and resources provided by the United States government, along with the testing agencies that were supporting the actual testing requirements.

The POSIX.3 standard is not a bad thing. It defines a methodology by which test methods and results of test cases written to these methods can be uniformly described.

If you are creating a standard it's a useful tool to ask yourself "how would I test this functionality or feature" as you write the specification. It ensures you read and possibly rewrite the specification properly. You may wish to deliberately not be complete in the definition, but these areas in a standard specification should be intentional.

This "testing" tool has even been proven. Several working groups have written test methods for their specifications, with some help from people historically involved in the original POSIX.3 effort. Many of these POSIX.3 members have formed the Steering Committee on Conformance Testing (SCCT) that oversees how test methods are applied and created in the working groups. The SCCT has been too busy to review these test methods in depth, but without judging whether the new test methods are good or bad, the working groups that have gone to the trouble of creating them have all felt that their base specifications are better defined for the effort. It seems that the tool works!

Now for the problem. Some time ago, the SCCT recommended to the Sponsor Executive Commit-

tee (SEC) that all POSIX standards must have associated test methods. These test methods would be standards as well. They convinced the SEC to make this a requirement.

Now, a standard cannot officially exit balloting without having a test method specification that is also a standard. This instantly sets up a directly competing body of text to the original standard. This is not a competing functional standard a la IEEE 802.n LAN standards. This is a competing body of text. (Note: ALL discussions of formal testing languages and formal specifications are red herrings here. Anyone wishing to hear my three Canadian cents worth on the subject can email me.)

Test methods standards will become the annointed specification for the test suite to demonstrate conformance by organisations with the funds or market presence to demand as much. Implementations can hit the narrower mark of the test suite (embodying the standard test methods) to naively certify rather than hit the standard itself. If you don't realise the subtle and nasty differences that can appear, spend some time with the POSIX.1 standard (IEEE Std 1003.1-1990), and with its newly declared standard test methods (IEEE Std 1003.3.1-1992).

And what happens when there are holes in the test methods? Some things cannot be tested. The standard still has requirements on these areas of behaviour, but they may not translate nicely. And there are some places where the test methods simply aren't complete. A reasonably recent draft of the POSIX.3.1 test methods had test methods for the POSIX.1 environment variables required by U.S. FIPS PUB 151-1 (the U.S. government profile of POSIX.1), but none for the other environment variables. The international community might wish to take note of this oversight on all LC_ environment variables, should the POSIX.3.1 standard get to ISO. What other holes are there?

There is a terrible balloting problem. Balloting apathy or overload is striking many places. The test methods documents are as big as the standards they repeat. Fewer people care about the test methods, they've seen the original specification and the job is done, right? We run the terrible risk of passing bad test methods documents if these documents are quickly processed through balloting groups whose members have little time on their hands. In the current commercial climate for standards, this is dangerous.

Then, of course, there is the maintenance problem. All useful standards have the same problem as all useful software. They need to be main-

tained. It's just slower and more tedious. A level of complexity has been added to the administration of the interpretations.

POSIX.1 has the fun little contradiction that `PATH_MAX` is the length of the pathname both explicitly including and excluding the terminating null byte. An interpretation was requested, and came back that it was an inconsistency and that both can be right.² Now what happens when someone requests an interpretation of a standard with its test methods?

If the request is leveled against the base, what guarantees are there that the test methods, i.e., a separate standard, will be kept synchronized? If it's against an inconsistency between the base and its test method standard, which one wins? If the `PATH_MAX` argument holds, then both are correct. Since one of them is implemented as a test suite to demonstrate conformance, which one wins in the real world?

Do test methods need to be standards? Who wins by forcing working groups to completely re-specify their work as test methods? Testing is expensive, but the market ultimately protects itself. What has been done in the TCP/IP space? (If you don't think TCP/IP is a successful widely implemented specification, stop reading now.) What about the C language? No one specified a set of test methods for the ANSI C standard. People in the know wanted to see how to test the C standard, and through a lot of hard work built the Plum-Hall test suite. The U.S. government created a FIPS for C and chose an available suite. There were no test methods for this work. No added burden on the volunteer standards community to respecify itself.

A great tool; but only a tool!

LIS – The Great Experiment

Language Independent Specification (LIS) is burden Number #2 on working group members. Two working groups have been operating in the POSIX space for quite some time in programming languages other than C. One is the POSIX.5 Ada Bindings group, which has re-cast the POSIX.1 standard into Ada, and is now working on POSIX.4 (Real-time Extensions). The second is POSIX.9 which has similarly cast POSIX.1 into FORTRAN 77, and is now considering what to do with Fortran 90. The two groups have finished their work. Two real standards exist within the IEEE standards realm:

IEEE Std 1003.5-1992 (Ada Bindings to IEEE Std 1003.1-1990.)

IEEE Std 1003.9-1992 (F77 Bindings to IEEE Std 1003.1-1990.)

A small digression is required on ISO POSIX. Along the way, IEEE POSIX entered the international community and an ISO Working Group (WG15) was created as its home in the Subcommittee on Programming Languages (SC22). WG15 is not a standards development group per se, in that it does no drafting of specifications. Its job is to review the draft IEEE documents and make recommendations to the IEEE, through the ANSI sponsored U.S. Technical Advisory Group (TAG) on POSIX, back to the POSIX Sponsor Executive Committee.

Do not be fooled. There is a substantial overlap in the key personnel of the IEEE working groups and people sitting in the WG15 meetings as individual technical specialists from their respective national POSIX standards groups.

ISO began trying to specify programming interface standards in programming language independent ways, such that the functional specification appears once, with multiple bindings. It seems expensive to continually re-specify a standard from one language into a standard in another language. There is the feeling that there is twice the work effort, plus the coordination effort.

A different international group, WG11, is working at defining abstract data types and such. All programmatic interfaces could eventually be described in some abstract functional way and each individual language binding would just "fall out" once the mapping from the abstract types to program language types had been established. Because of early experiments in specifying standards this way, language independence was inflicted on POSIX as a requirement from WG15. POSIX the Guinea Pig. WG11 had never been faced with POSIX.

All this means every standard becomes two standards. There is a book describing the functional specification in abstract data types, and a book specifying a mapping to a real programming language's syntax, along with additional required semantics. Try re-reading each of the last few paragraphs, and after each repeat, "It is intended to be used by both application developers and system implementors." Ideally, ISO WG members believed that the functional specification would be a "thick" book, and that the language binding would be "thin."

2. IEEE Std 1003.1-1988/INT, 1992 Edition, Interpretation Number: 15, p. 36.

The Ada group, POSIX.5, chose not to split their work. They argued it was too late in their project and that a sufficiently mature POSIX.1 LIS did not exist. They further argued that they had to produce a “thick” language binding reproducing much of the semantic content of the POSIX.1 book, recast into Ada-speak, in-line. Programmer usability was very high on their list of priorities. Think about that for a minute.

I work in an environment where we regularly refer to the POSIX.1 standard. We write code that needs to be portable to many non-Unix based architectures that provide POSIX.1 interfaces. All of our many copies of POSIX.1 are very dog-eared and marked up. We use our copies daily. It is a useful book from which to program. It is not a tutorial. It is a programmer’s reference.

I recently had to go through the POSIX.5 and POSIX.9 standards. I am not an Ada programmer, but still found the information I needed to find, in an easily understandable form. The POSIX.5 group did their job well. Yes, it is a thick binding repeating the semantic functional material of POSIX.1. And yes, even though the POSIX.5 standard is supposed to exactly mirror the POSIX.1 standard, I found a bug (or at least something about which to request an interpretation). But I found the information, clearly laid out; even the bug!

The POSIX.9 (FORTRAN 77) working group chose to attempt a thin language binding to POSIX.1. They were very tight for resources and they wanted to do the right thing with respect to the ISO WG15 requirements. Through no fault of their efforts, I found it to be a difficult book to use, and I was a Fortran programmer in a previous incarnation.

First, you immediately run into the two book issue. Look up the syntax in POSIX.9 which immediately punts you to the semantics in POSIX.1. So you jockey about two books in your lap, continually cross referencing.

Second, you continually switch frames of reference. In one book, there is a solid real world line of language syntax; in the other book, a description of that syntax’s semantics in a different specification language (C).

In balloting the POSIX.1 Language Independent Specification (LIS), I ran into the same problems. Two books, two frames of reference. At least POSIX.1 Classic (IEEE Std 1003.1-1990 == ISO/IEC IS 9945.1:1990) stands as an existing reference against which to compare these models. When we begin balloting drafts of API standards as LIS

and attendant bindings in at least one language, will we be able to catch all the holes?

The IEEE paid to have the initial drafts of POSIX.1 LIS and its C binding (POSIX.16) produced. They couldn’t get the work done any other way. Paul Rabin worked long and hard to produce guidelines for writing LIS and language bindings. This work was done within the IEEE POSIX realm, although Paul liaised closely with ISO WG11 and WG15. The few IEEE POSIX working groups that have attempted partial or complete drafts of their work using these guidelines, have immediately started finding problems in their previous C language specific descriptions. Just like test methods, prodding the text by attempting to recast it into a different form made a better specification.

Again, one has to ask if this is a good way to define standards. A tool to test the specification, yes. The specification itself? One has to assume that the standard has an audience, and that usability is an important factor. One should assume that the standard is based on existing practice for the most part. That existing practice is in a particular programming language for API type standards. Those will be the first people to come forward to develop the standard. (There has to be a need to standardize.)

If others with a different programming language background participate, this would be ideal. If the experience with the functionality exists in more than one language, and they all want to come to the table, this is even better. But we do not live in an ideal world. Specifying the functionality in a hard to use (2 document/2 context) format is error prone, especially when the document is being balloted. Until formal methods become a common method of expression, we are stuck with English descriptions, and the exacting programming language syntax of the existing body of experience in that area of functionality.

Language Independent Test Methods

Yes, you read the title correctly. If the functionality can be abstracted, described exactly, then bound in various programming language syntaxes, so to can the test methods of that functionality. Think about how you would test an Ada run-time implementation of POSIX.1.

And each is a standard. So there is a base programming language independent functional specification (LIS) standard, a programming language binding standard, the LIS test methods standard, and the language binding standard for those test methods. Balloting will kill us. We will produce unusable junk if we continue.

Simple economics says we're doomed. The IEEE is being forced to pay up into ANSI for its international standards efforts. To cover the costs of simply balloting the quantity of paper, the IEEE has been forced to start charging \$25 US to join balloting groups. To cover the international participation, they've considered raising this to \$50 US. That means it will cost the individual professional programming member of the IEEE \$200 to join the balloting groups for a set of standards that represent a simple piece of functionality in which they are interested.

One might argue that a programmer will only join two balloting groups, for the LIS and language binding. Because the test methods (LIS and language binding) are a competing body of text, however, they will need to check the test methods to confirm they are accurate. Because of government procurement policies here and abroad, the test methods will be important!

An Architect's Nightmare

LIS, language bindings, LIS test methods, and their bindings. Now imagine that we start amending the four standards at once. POSIX.6 (Security Extensions to POSIX.1 and POSIX.2) will amend POSIX.1 and POSIX.2 somehow at some point in the not too distant future. So will POSIX.4 (Real-time Extensions), POSIX.8 (Transparent File Access), and POSIX.12 (Sockets/XTI).

The original POSIX.6 document, which did contain all the information they could put together on POSIX security has just needed to be split SIX ways:

- The API as an LIS, to amend POSIX.1/LIS,
- The API as a C-binding, to amend POSIX.16,
- The API test methods in LIS form, to amend POSIX.3.1 (which currently isn't in LIS form),
- The API test methods as a C-binding, to amend POSIX.3.1 (in its current C form?),
- The utilities, to amend POSIX.2,
- The utility test methods, to amend POSIX.3.2.

Can't wait.

The Problem Revisited

If POSIX continues on its current course, one of two things will happen.

ONE – They will succeed. The useful standards which do exist will be amended to a user unfriendly form. An ugly unusable set of stan-

dards will eventually be born. Because of the lack of use, they will fail. People will not use them. It will be too easy to ignore them. Programmers will not be able to rely on a certain portability model. The vendors will continue to sell completely proprietary implementations.

TWO – They will fail. Under its own weight, it will collapse. If not with a bang, then with a slow sickening crunching sound. The people with the knowledge will get tired, or lose support (as they obviously aren't producing anything to show their management in recessionary times). POSIX.1 will become unusable as it is amended and amended and almost amended. ("If we wait for another 6 months, we'll be able to get all the wizzy features in POSIX.42....")

ONE AND A HALF – Life isn't this black or white. The ugly truth will lay in the middle. We're talking about several thousands of pages of functional specification. We're talking several hundred people in working groups, plus hundreds more in balloting groups, plus the unsuspecting time-delayed purchasing public. The death will be long and painful. Senility will set in first.

Solutions!

OK. Let's stop the gloom and doom. Let's take an optimistic pro-active view! What to do about the problems of POSIX? Let's put it on a diet.

Remove the continued requirement on balloting the test methods as standards. The Steering Committee of Conformance testing would no longer have a function. Its members could go do real work in the POSIX.3 update effort, adding to a useful document which provides a tool for testing the specifications developed in working groups.

These working groups would immediately cease worrying about developing complete test methods documents. Those that cared, would, when occasionally confronted with ugly passages in their drafts, have a useful tool (POSIX.3) to use to try answering the question, "how would I test this?"

Ballot groups could concentrate on the real specification in front of them. Repeat again: Bad test methods standards will be dangerous in the marketplace.

Individual technical members in working groups could stop worrying about completely re-specifying their document. Possibly some that cared, with the newly found time, might actually write

some real honest-to-god test cases. These would surface, instead of everyone waiting to see which way the testing wind was going to blow by large governmental agencies here and abroad. These test cases might even be used, therefore useful.

Should these large governmental testing concerns wish to compare the merits of test suites, they could require that they are documented, and record results according to POSIX.3. Render unto the standards community that which is the standards community's, and render unto the marketplace that which is the marketplace's.

Who can act on this recommendation? The IEEE POSIX Sponsor Executive Committee can. They are made up of the working group chairs, the steering committee chairs, and institutional representatives. There is a list of these at the end of the article, with email addresses. Send them e-mail. It really only takes a minute. It will save you a lot of future grief to take the minute to ask questions NOW!

There is also a list of some important heads of delegations within the ISO POSIX WG15. WG15 is considering forwarding IEEE test methods documents as standards at the international level. Then we can all live with any mistakes in the U.S. government procurement policies! E-mail soon! E-mail often!

Let's continue the POSIX diet. Programming Language Independent Specifications should be stopped for the time being. The IEEE has put forward an incredible good faith attempt. The experiment should be considered a success! We have demonstrated that we don't yet know enough about specifying API standards in this abstract way. We should cease to hold up the working process.

Once the problem is better understood, and our methods of describing things in an LIS improve, we can begin exploring the possibilities. Notice that I didn't say retrofit or recast. I said explore the possibilities. Until we actually add a few of the large amendments to the base standard, changing its format midstream just opens things up for abuse and error. Let's do it a few times in languages that many of us understand, i.e. C, Fortran, Ada, before tackling the problem with little understood methods, which have been untried at this scale.

What would happen? Working groups would spend less time trying to recast their work (again!) into LIS. They would spend more time on the real specification, making it usable "for application developers and systems implementors."

When the existing working groups want to bind something in more than one language, they arrange to attend one another's meetings, and they work together. This sometimes takes the form of the complex strained negotiations that are the consensus process. This process is already in place in POSIX and has been for some time. It works. The LIS has not been required in producing the usable standards documents to date.

Who can act on this recommendation? Once again, the IEEE POSIX Sponsor Executive Committee can. This one is harder, however, as ISO WG15 is also involved.

First, the SEC has to be willing to say "no". This is not a surly uncooperative "no". A huge work effort has gone into the LIS experiment. There is real experience in the IEEE POSIX projects with this. The SEC can say "no" with confidence based on experience. ISO cannot claim the same experience. (If they could, they would have been helping us a long time ago.)

Second, ISO WG15 has to be willing to say "no." Remember that there is a sizable overlap in the small membership of WG15, and members of the SEC. The IEEE POSIX working groups have many international members who show up in the Canadian, UK, American, and German delegations. Education is certainly not the problem here, however, communication might be.

Other special working groups within ISO may be concerned with this approach, but again the experience lies within the IEEE POSIX working groups, which overlap with ISO WG15. Other ISO concerns should be acknowledged and put to rest. Once again I say: E-mail soon! E-mail often!

Ultimately, in a worst case scenario some level within ISO could refuse to accept IEEE POSIX drafts for ISO balloting. I believe even this case should not be of concern, based on the following examples:

ISO WG15 has not accepted the perfectly useful IEEE POSIX.5 for international standardization, since it did not fit the ISO requirements. ISO WG9 (ISO Ada Working Group) has been very concerned by this action and is attempting to fast track the IEEE POSIX document.

A representative from AFNOR (France's National standards organization) voiced strong support for the IEEE POSIX groups to continue to bring forward the standards as LIS at the last ISO WG15 meeting. He then immediately expressed grave concerns that POSIX.4 be brought forward

as quickly as possible in its current C-based form to the Draft International Standard (DIS) state. You see, the French government can procure against a DIS.

Ultimately, if the IEEE POSIX working groups do their job right and produce useful and usable standards, the market will demand their use, even if they have to be fast-tracked into the back door to make them international standards for the international market place. Twisting the standardization process away from defining detailed

specifications towards suiting procurement processes from organizations that are too big to change is wrong!

POSIX has market momentum. It will affect the way you do things. The working groups have produced useful standards, but that is now in jeopardy. You can affect the process. If you can't get directly involved, e-mail the appropriate people below and ask questions! Explain your concerns! Otherwise, you'll have to live with their decisions.

Who Ya Gonna Call?

Position	Name	E-mail
IEEE Concerns		
Chair SEC	Jim Isaak	isaak@decvax.dec.com
Vice Chair Interpretations	Andrew Twigger	att@root.co.uk
Vice Chair Balloting	Lorraine Kevra	l.kevra@att.com
Chair Steering Committee on Conf Testing	Roger Martin	rmartin@swe.ncsl.nist.gov
Chair Project Management Committee	Shane McCarron	s.mccarron@ui.org
Chair POSIX.1	Paul Rabin	rabin@osf.org
Chair POSIX.2	Hal Jespersen	hlj@posix.com
Chair POSIX.3	Lowell Johnson	3lgj@rsvl.unisys.com
Chair POSIX.4	Bill Corwin	wmc@littlei.intel.com
Chair POSIX.5	Jim Lonjers	lonjers@prc.unisys.com
Chair POSIX.6	Ron Elliot	elliott%aixsm@uunet.uu.net
Chair POSIX.7	Martin Kirk	m.kirk@xopen.co.uk
Chair POSIX.8	Jason Zions	jason@cnd.hp.com
Chair POSIX.9	Michael Hannah	mjhanna@sandia.gov
Chair POSIX.12	Bob Durst	durst@mitre.org
USENIX Institutional Rep	Jeff Haemer	jsh@canary.com
EurOpen IR	Stephen Walli	stephe@mks.com
Uniforum IR	Ralph Barker	ralph@uniforum.org
DECUS IR	Loren Buhle	buhle@xrt.upenn.edu
OSF IR	John Morris	jsm@osf.org
Unix International IR	Shane McCarron	s.mccarron@ui.org
X/Open IR	Derek Kaufman	d.kaufman@xopen.co.uk
WG15 Concerns		
Convenor WG15	Jim Isaak	isaak@decvax.dec.com
US Head of Delegation	John Hill	hill@prc.unisys.com
Canadian HoD	Arnie Powell	arniep@canvm2.vnet.ibm.com
UK HoD	David Cannon	cannon@exeter.ac.uk
German HoD	Ron Elliot	elliott%aixsm@uunet.uu.net
Dutch HoD	Herman Wegenaar	(phone: +31 50 637052)
Japanese HoD	Yasushi Nakahara	ynk@ome.toshiba.co.jp
French HoD	Jean-Michel Cornu	jean-michel.cornu@afuu.fr
Danish HoD	Keld Simenson	keld@dkuug.dk

Report on POSIX.0: The POSIX Guide

Kevin Lewis <klewis@gucci.enet.dec.com> reports on the October 19-23, 1992 meeting in Utrecht, NL

The ballot submission period for POSIX.0 closed on September 15, 1992. Below are the ballot statistics:

86 ballot group individuals
81 ballot group formal members
69 ballots submitted = 85% returned
11 abstentions
30 negative
28 affirmative = 48% returned (16 affirmative w/ no comments)
1127 comments/objections (approximate)

POSIX.0 dedicated all of the October meeting towards ballot resolution. The section leaders are serving as the technical reviewers for ballot resolution. They received 30 ballots via e-mail approximately three weeks prior to the meeting. (Three of the 30 were from individuals not on the ballot list. The group decided to treat them as "parties of interest.") Fifteen were received by the ballot coordinator on the Friday before the meeting, so the technical reviewers saw these for the first time in Utrecht.

The group focused on identifying those objections felt to be substantive, key, or "show stoppers." The areas that these fell into include profiles, the reference model, and public specifications.

Let me note at this point that just about everyone in the group, including Yours Truly, demonstrated a clear case of memory shutdown, i.e., forgetting how we dealt with process and disposition issues during mock ballot. I attribute that to this last quarter requiring no working group activity aside from individuals' submitting their ballots. So it took the group about a day to "reboot."

In parallel, the guide is also in the review and comment process within WG15 and SC22. As of this writing, no comments have yet been received.

The TCOS SEC approved a resolution to forward the next draft of the guide, which will be the first recirculation draft, to SC22 for CD registration.

The group established the goal of completing ballot resolution within 7-10 days after the January

93 meeting. A tentative first recirculation meeting has been identified within the April 1993 time frame. This will be confirmed before the January meeting.

Overall, the guide is in good shape. The big question, implicit as it may be, is how well we will fare beyond the 75% requirement for affirmative votes before the guide can be published. It is too early to say. I'll have a much better feel after the January meeting.

Report on POSIX.2: Shell and Utilities

David Rowley <david@mks.com> reports on the October 19-23 meeting in Utrecht, NL

Summary

The grand moment has arrived, we have a final POSIX.2 Standard:

IEEE Std 1003.2-1992

Approved by the IEEE Standards Board on September the 17, 1992, POSIX.2-1992 is the culmination of over five years of the working group's efforts. The standard consists of both the "Dot 2 Classic" and "Dot 2a" components, previously balloted as separate standards. The IEEE Standard (based on the new Draft 12) is identical (at least from a technical standpoint) to the draft ISO standard, ISO/IEC DIS 9945-2:1992.

NIST continues to work on the draft of a new FIPS (Federal Information Processing Standard) for POSIX.2, expected in final form by early 1993.

POSIX.2b work continues to proceed, incorporating symbolic link support within a number of utilities, a new PAX archive format, and addresses a number of international concerns regarding locales. The PAX format is still based on the old but standard ISO 1001 tape format.

Test assertion work nears completion. The POSIX.2 assertions have almost full coverage, and will go to ballot again in December. The POSIX.2a test assertion work is going well, with almost all assertions complete, including vi. These will be folded in to the next draft of the POSIX.2 test assertions.

The test assertion work for POSIX.2 will be renamed P2003.2 instead of the current P1003.3.2.

Background

A brief POSIX.2 project description:

- The base utilities of the POSIX.2 standard deal with the basic shell programming language and a set of utilities required for the portability of shell scripts. It excludes most features that might be considered interactive. POSIX.2 also standardizes command-line and function interfaces related to certain POSIX.2 utilities (e.g., `popen()`, regular expressions, etc.). This part of POSIX.2, which was developed first, is sometimes known as "Dot 2 Classic."
- The User Portability Utilities Option or UPUO, is an option in the base standard (previously known as POSIX.2a). It standardizes commands, such as `vi`, that might not appear in shell scripts, but are important enough that users must learn them on any real system.
- Some utilities have both interactive and non-interactive features. In such cases, the UPUO defines extensions from the base POSIX.2 utility. Features used both interactively and in scripts tend to be defined in the base utility.
- POSIX.2b is a project which covers extensions and new requests from other groups, such as a new file format for PAX and extensions for symbolic links. It also includes resolution of items arising from comments by ISO Working Group 15.

POSIX.2 is equivalent to the International Standards Organization's ISO DIS 9945-2 – the second volume of the proposed ISO three-volume POSIX standard.

Publishing

Now that the Standard has been approved by the IEEE, everyone is anxiously awaiting the final published volumes. They will be printed on A4 paper in two volumes: the core standard (Sections 1-7), and the annexes. The set should be available from the IEEE sometime in the March 1993 time frame at a total page count of around 1300 pages.

POSIX.2 FIPS

NIST (National Institute of Standards and Technology) is still preparing the draft FIPS (Federal Information Processing Standard) for POSIX.2. The goal of the FIPS is to directly adopt, rather than adapt, POSIX.2 as a procurement standard. The selection of options and extensions will be left to the procurement officer. This will lead to even greater use of the standard, due to the flexibility this offers the agencies wishing to reference POSIX.2.

NIST Draft Request for Test Technology

NIST has issued a draft of a Request for Test Technology. NIST is seeking candidate test suites from which to select one test suite to measure conformance to the proposed POSIX.2 FIPS. It must be based on TET (Test Environment Toolkit from OSF-UI-X/Open), cover all assertions from POSIX.3.2, and satisfy the general test method requirements specified in POSIX.3. The suite must also be commercially available (either now or in the future). The full RFTT is due out early in the new year.

X/Open Request for Proposal

X/Open is in the final stages of signing the contract for the Integrator they have chosen for their combined POSIX.2/XPG4 Commands and Utilities test suite, to be integrated into a future release of VSX (Validation Suite for XPG). The Integrator will likely be publicly announced before the end of the year. Work is to start early in 1993, and should result in a suite being publicly available early in 1994.

Test Assertion Group Name Change

The IEEE is in the process of renaming the test suite working groups to a parallel numbering system to P1003. As of March 1993, the POSIX.2 test methods work will be numbered P2003.2. This should ease the confusion of many similar sounding working groups containing numerous dots and digits.

The ballot for Draft 8 of the POSIX.2 test assertions starts December 6th and ends January 6th. Some ballot resolution will be attempted at the January POSIX in New Orleans (the 11th to the 15th). Draft 8 includes assertions for all utilities except those from Section 5 of POSIX.2 (the User Portability Utilities Option, formerly POSIX.2a). These missing assertions will be included for the full re-ballot, Draft 9, expected sometime in March 1993.

POSIX.2b

The current draft of POSIX.2b, Draft 4 – August 1992, includes a number of extensions and additional utilities. The BASE64 encoding from MIME (Multipurpose Internet Mail Extensions, RFC 1341) has been incorporated into `uuencode/``uudecode`. The "iconv" utility for character set conversion has been added from XPG4. Print field widths have been added to the "date" command. Support for symbolic links has also been added to a number of utilities.

Locales

A proposal from Thomas Plum regarding a new locale specification format from P. J. Plauger was discussed. Although the format has some interesting features, the codeset specific nature of the format limits its usefulness, and was deemed dangerous in a POSIX environment. A liaison statement to WG14(C), WG20 (Internationalization) and WG21 (C++) will be drafted by the Chair.

Yoichi Suehiro (DEC Japan) made a proposal to extend LC_TYPE to handle user-definable character conversions and user-definable character classes. These were both felt not to be within the scope of POSIX.2, but may be reconsidered at a later date.

Extensions to LC_TYPE were approved to specify the display/print widths of characters in the locale. This information will be specified by using the keywords "width1", "width2", etc. There will also be a "default_width" keyword which specifies the default width occupied by all characters not specifically mentioned in one of the "width" classes.

"era_d_t_fmt" had accidentally been left out of the LC_CTIME category. This will be corrected through POSIX.2b.

There was a long discussion on multibyte and stateful encodings and the need for coordination between ISO 9945-1 and ISO 9945-2. This will be discussed further in subsequent meetings.

New PAX File Format

The request for alternate PAX format proposals generated only a few pointers to other file formats, particularly the MIME standard (RFC 1341). Mark Brown has volunteered to write up a rough draft of a MIME-based PAX format to be discussed in New Orleans. Other than that, the group continues to work with ISO 1001. The group has also agreed to adopt Gary Miller's (IBM Austin) new File System Safe UTF (UCS Transformation Format) which specifically stays away from the codepoints representing the ASCII "/" character and null bytes.

Character set conversions issues within the PAX format can now be handled in a generic, system-wide manner given that the "iconv" utility has been added to the standard. This should result in a much more useful and consistent system for the user.

Report on POSIX.4, POSIX.4a, POSIX.4b, POSIX.13 (Real-time POSIX)

Bill O. Gallmeister <bog@lynx.com> reports on the October 19-23, 1992 meeting in Utrecht, NL

Summary

Well, for all those of you who've been breathlessly following the progress of the real-time POSIX proposals these last few months, you may have noticed a dearth of USENIX updates on the subject. Blame the snitch. He's a slug, and forgot to do the last report. This report will cover the last two meetings – July (Chicago) and October (Utrecht).

The real-time working groups are making quiet, steady progress on POSIX.4 and POSIX.13, which are two of our proposals that are out to ballot. In fact, we fully expect to turn POSIX.4 into a real live standard on or about January, 1993. (It depends more on when the high muckety-mucks of IEEE get around to it than on anything else, in my opinion.)

POSIX.13 is our profile document, which calls out what parts of POSIX you need in order to run POSIX on your Cray or your cruise missile, depending on what you may have. The situation with POSIX.13 is really pretty interesting, so we'll end with that to give you something to look forward to.

Rounding out our picture, we have POSIX.4a – threads – which seems to have completely vanished into the hands of the technical editors. Those of us who actually would like a useful threads standard sometime in this century are getting a little impatient. We have rather little recourse, however, since documents in ballot are not really the province of the working group anymore. Threads is a grown-up standard now and it'll just have to look out for itself.

And, finally, the Yet More Real-Time additions in POSIX.4b are proceeding apace in the working group.

POSIX.4: Real-Time Basics

Good news here. POSIX.4 is actually approaching finalization! After a couple of changes that had us a little worried (the addition of `mmap()`, and the change to semaphores from binary to counting), we found the balloting group basically agreed whole-heartedly with the way things were going.

That's not to say they didn't have plenty of other things to kvetch about, but then that's what bal-loters are for.

But at this point, we have passed Draft 13 through a recirculation, and from what I am told, the initial results look quite promising. Basically, very little of the POSIX.4 document is open to comment at this point, and the next circulation should be small, fast, and quickly resolved. That done, we can take POSIX.4 to the IEEE standards board at their June meeting. It is already in the Committee Document registration phase at the ISO WG15 level, on its way to international standardization.

POSIX.4 is one of the last standards that was allowed to pass without a language-independent specification and test methods. One of our next jobs is to produce a version of POSIX.4 in LI form, with test methods. A group of volunteers has been formed to start on that work, and should have some progress to report at the January meeting (but not much, given the holidays between now and then).

POSIX.4a: (The Long-Lost) Threads

What's going on with threads? Don't ask us. We're just the working group. As far as I've been able to tell, everyone involved in moving the threads chapters through their ballot has either lost interest, had children, gotten out of school and started making the big bucks, moved to France, or been involved up to their eyeballs in justifying their own continued existence at their various companies.

I'm told that threads needs to be kick-started a little bit. In Utrecht, we had a serious contingent of angry natives wanting to know what was up with threads. My prediction (and take it for what it's worth) is that the threads technical reviewers have until the January meeting to make some visible progress on their standard, or we might get some new technical reviewers who are less strapped for time.

POSIX.4b: Extra Real-Time Interfaces

This is a proposal that not many people know too much about, so I'll give a fast introduction to it. POSIX.4 was started to extend POSIX.1 for real-time. POSIX.4 settled on a subset of functionality for real-time – things we thought were absolutely crucial, and most importantly, things we could actually make some progress on. The more contentious items were left behind for a “future standardization” effort. That effort is POSIX.4b.

The facilities of POSIX.4b are more esoteric and less widely applicable, although they are absolutely essential for certain real-time applications. POSIX.4b has chapters for:

- direct application access to interrupts,
- device control (a.k.a. *ioctl()*, although we had to change the name to protect the existing),
- *spawn()* (a combined fork-and-exec which can be more easily performed than fork/exec on an MMU-less architecture),
- Sporadic Server scheduling (a scheduling discipline used in conjunction with Rate Monotonic Analysis to support, fittingly enough, sporadically-interrupting devices and other things that take unpredictable amounts of time),
- and CPU time monitoring (the POSIX.4 version of *times()*, essentially allowing one thread to monitor the execution time of another).

There is also work ongoing on extended memory management, something to allow one to allocate from distinct, special “pools” of address space (memory attached to a particular bus or device, in particular.) This chapter is up in the air and might go away.

The POSIX.4b proposal is proceeding along rather fast. It's a little terrifying to see a proposal that aims to allow an application to manhandle an interrupt vector, coming at you full speed ahead. Luckily, we have the (I hesitate to say it) stabilizing influence of people from POSIX.1 (who are interested in *spawn*) and sundry large, entrenched camps of UNIX aficionados in the group on an intermittent basis. Hopefully this influence will help produce something that is appropriate for standardization. It would certainly help, in my opinion, if more mainstream UNIX types were to give us a hand at UNIX-ifying the POSIX.4b proposal before it hits balloting. Maybe some of you nice people can drop in on the working group in New Orleans in January.

POSIX.13: Real-Time Profiles

This is the fun one.

POSIX.13 was the first profile proposal to hit balloting. We played by the rules. We produced our document. We formed our balloting group. We went to ballot. We got substantial approval, enough that very little of POSIX.13 should be open to comment on the next recirculation.

Oh, did I mention how POSIX.13 breaks just about every rule of how a profile document should be built? This unfortunate fact has led to some hand-wringing among the POSIX powers-that-be. The Powers would probably like for POSIX.13 to withdraw itself from ballot (despite the fact that it's mostly approved by the balloting group) and just go away until it can be reformed as a good POSIX citizen.

What are POSIX.13's crimes? Well, it's four profiles, not one. That's a problem, but not a big one. We could split the document with only minimal impact on the Spotted Owl population (and the lumberjacks would love us).

A bigger problem is that POSIX.13 calls for subsets of POSIX.1. Like, a POSIX without the ability to *fork()* (can't do it on an embedded, MMUless target), or *exit* (what sense does that make if you can't *fork()*?).

The smaller profiles of POSIX.13 are undoubtedly useful to people building embedded applications, however, there's a lot of consternation that something without a small modicum of UNIX-ness could possibly be allowed to call itself POSIX. So, lately, compromise wording was adopted in the committee whose job it is to make rules about profiles. That wording would allow the minimal profiles to be called "Authorized POSIX Subset Standardized Profiles," whereas something with a real POSIX.1 would be called a "POSIX System." And, of course, we would still need to convince POSIX.1 to subset itself.

Meanwhile, the POSIX.13 proposed standards are in the hands of – gasp! – people who are interested in doing real work. And it is clear that POSIX.13 would be useful for those doing real work, even if it is confusing and nasty by POSIX standards. [ed. – *Nasty pun, Bill.*]

I predict we'll see an essentially-approved version of POSIX.13 in a year, which will then have to wait for POSIX.4a to be finalized before the profiles really mean anything (you can't call out threads support when there is no threads standard). I further predict that the POSIX powers that be will declare POSIX.13 out-of-bounds, and that people will continue to use POSIX.13 anyway.

Report on POSIX.7b: Software Administration

Esti Koen <emk@cray.com> reports on the October 19-23, 1992 meeting in Utrecht, NL

I attended the POSIX.7b meeting in Utrecht, never having been previously exposed to POSIX. Lacking the historical perspective, it was difficult for me to identify when the discussion was a clarifi-

cation of an already agreed upon point versus a major shift in emphasis or direction. If this report seems somewhat lacking in detail or introductory, it reflects my own level of involvement to date.

For the purpose of this report, I assume readers are mainly interested in broad decisions concerning the content of the standard or a shift in direction and expected balloting dates.

Early attempts to standardize the nonexistent "common practice" of software administration seemed doomed to failure. (I don't envy those early pioneers.) POSIX.7 finally adopted the network view of a managed system. Forging ahead in areas where they feel they can make consensus based progress, POSIX.7 is now split into two documents called POSIX.7a (print queue administration) and POSIX.7b (software administration).

Recognizing the need for information describing existing practice in the area of network wide system management, the Open Software Foundation (OSF) solicited technologies from industry that could be integrated to simplify system management in heterogeneous computing environments. In October, 1991, OSF announced that they had chosen Hewlett Packard's Software Distribution Utilities to provide the basis for the OSF Distributed Management Environment (DME). The current draft of POSIX.7b is a roughly one year old descendant of the External Specification that describes the HP Software Distribution Utilities.

The original HP implementation suggested an object orientation but it was not developed using a rigorous object oriented specification language. In one year of POSIX meetings the group has made significant progress in further defining the attributes of the managed objects, but the specification is still incomplete and at times ambiguous. There is much discussion concerning object behavior.

Open issues include the question of allowing multiple Management Information Bases (MIB), and which attributes of a software object can be used, and how they are used as a selection mechanism.

Although invention by a standards committee is not advisable, it seems unavoidable when the base design is incomplete for the purposes of the standard.

Several decisions regarding general content were finalized. There will be no API included in the standard. An informative annex which provides information on how one implementation communicates between the manager, source, and tar-

get roles will be included. A rationale section which informs the reader as to the intent and history of the standard will also be included.

The serial media format was previously specified as `tar`, but will now be specified as being readable and writable by `pax` (POSIX.2-1992). Locking mechanisms are considered to be an implementation detail and outside the scope of the standard. A command line option will be provided to permit interaction sufficient to handle multi-volume media.

The group discussed rewriting part of the document using the ISO Guidelines for the Definition of Managed Objects (GDMO). The process of rewriting using GDMO would have the beneficial side effect of highlighting inconsistencies, omissions, and redundancies. In fact, it was advised that the draft would not be adopted by ISO unless GDMO was used.

The active participants did not embrace the idea wholeheartedly because a drastic structure change could further delay the balloting schedule. Mock ballot is planned to occur after the January meeting. Budget constraints may impose a time limit on the standards activity, and active participants fear having the POSIX.7b standards activity permanently interrupted before going to ballot. Refinement of the existing object definitions and behaviors continues at a fast pace.

Report on POSIX.14: Multiprocessor Profile

Rick Greer <rick@ivy.isc.com> reports on the October 19-23, 1992 meeting in Utrecht, NL

The big news in the POSIX.14 working group is that we have inherited the POSIX.18 draft from Donn Terry and are now responsible for seeing it through balloting. POSIX.18 is the Platform Environment Profile, more commonly known as a profile to describe the traditional multi-user Unix platform.

Having been assured that the POSIX.18 document was "practically ready for balloting," we traded POSIX.14's March 1993 balloting slot to POSIX.18. Remember that this year there are so many documents in ballot that a strict timetable is being used to control the potential administrative overload. Our document's ballot slot had been allocated as a purely defensive measure anyway – see below. We also decided to keep the balloting group open right up to the last minute, so those interested in paying \$25.00 for the privilege of complaining may still do so. [Ed.– *This may be raised to \$50.00 in the new year!*]

We made one major change to the POSIX.18 draft: The C language feature is now required. It had been optional. Our reasoning for this was two-fold. First, we realized that because there was no requirement that a given implementation provide a specific language feature, people could write POSIX.18 compliant applications that would not run on POSIX.18 compliant implementations! By requiring C at a minimum, vendors can guarantee portability of other languages, in particular FORTRAN and ADA, to all POSIX.18 compliant implementations by writing their runtime libraries in C.

Secondly, given that POSIX.18 is supposed to codify "classic UNIX," and since classic UNIX has always included a C compiler, albeit the "classic" K&R compiler, not `c89`, we felt it appropriate to require C language support in POSIX.18.

The working group also made a number of minor editorial changes to the document, mostly removing redundant text, which brought it down to less than half its original size!

As for POSIX.14's real purpose, the POSIX multiprocessor profile, we decided not to ballot the current draft after all. We had originally decided to put POSIX.14 out to ballot in March in an attempt to be in ballot by the time the Profile Steering Committee (PSC) finalized its rules for "Standard Posix Profiles." We reasoned that if profile groups that were in ballot at the time the rules were adopted were grandfathered in such a way as to allow them to ignore said rules, POSIX.14 might be the only profile to which the rules applied. This seemed a bit unfair.

It now appears, however, that all profiles will have to follow the PSC rules before they can come out of ballot.

So we're back to proposing new MP interfaces for POSIX.1 and POSIX.2 that would fill various semantic gaps in MP systems that will be noted in the POSIX.14 draft. This includes describing parallel behavior for a number of common utilities (e.g., `make`, `find`, `grep`, `xargs`,) as well as describing special MP features of system administration functions such as `ps(1)` and `times(2)`. We also continue to argue about processor binding: can we specify enough of this in an architecture-independent manner to make it worthwhile?

One interesting point made at the October meeting was that many of the participants in our working group feel that our major contribution will not be the MP profile, so much as our monitoring of other POSIX work to make sure that any new interfaces do not cause major headaches for

MP implementations (e.g., the work that we've already done with respect to pthreads). With this in mind, we have proposed a new name for the group: POSIX.14 – the POSIX reentrancy police!

Report on POSIX.17 - Directory Services API

Mark Hazzard <markh@rsol.unisys.com> reports on the October 19-23, 1992 meeting in Utrecht, the Netherlands

Summary

A recirculation ballot of Draft 4.0 of POSIX.17 completed just prior to the Utrecht meeting and the group met primarily as a ballot resolution team. All but one of the outstanding comments and objections were resolved.

The next draft (Draft 5.0) will contain editorial changes and two minor technical changes. The changes will require another recirculation ballot. Only the pages affected by the technical changes will be distributed and can be balloted upon.

We expect to produce a Draft 5.0, do the "mini" recirculation, process and incorporate changes (if any) in time for the March 1993 IEEE Revcom meeting. Given this schedule, you can expect publication of our approved specifications in the middle of 1993.

The US TAG to ISO/IEC JTC1 has stated their intention to forward our specification to ISO for fast tracking (direct ISO ballot) when approved as an ANSI/IEEE standard.

Introduction

The POSIX.17 group has generated and is currently balloting a user to directory services API (e.g., API to an X.500 DUA – Directory User Agent). We used APIA – X/Open's XDS specification as a basis for work. XDS is included in XPG4 and has been adopted as part of both OSF's DCE and UI's Atlas.

XDS is an object oriented interface and requires a companion specification (XOM) for object management. XOM is a stand-alone specification with general applicability beyond the API to directory services. It will be used by IEEE 1224.1 – X.400 API (and possibly other POSIX groups) and is being standardized by POSIX/TCOS as P1224. A draft of P1224 is already in ballot.

POSIX.17 is one of five "networking" groups that currently make up the IEEE TCOS/POSIX Distributed Services and as such, POSIX.17 comes

under the purview of the Distributed Services Steering Committee (DSSC).

Status

Draft 4.0 of POSIX.17, which included all the technical, editorial, and format changes identified in the July Chicago meeting, completed a recirculation ballot prior to the Utrecht meeting. POSIX.17 was recirculated as four separate specifications:

- P1224.2 Directory Services API – Language Independent Specification
- P1326.2 Test Methods for P1224.2
- P1327.2 C Language Binding for P1224.2
- P1328.2 Test Methods for P1327.2

NOTE: During a special ad hoc meeting of the US TAG to JTC1, POSIX.17 was one of three TCOS APIs recommended for fast track to ISO. In order to accommodate the ISO format, POSIX.17 was required to be split into four separate parts (documents), hence the four specifications.

The group spent a majority of the meeting processing the results of that ballot and planning for another "mini" recirculation and final submission to IEEE RevCom for approval. Most of the comments were editorial in nature. However, two minor technical corrections were suggested and accepted by the committee, which (in the opinion of the IEEE) will require another (mini) recirculation.

All but one of the outstanding comments and objections were resolved for Draft 4.0. These results exceed the level of consensus (75%) required by the IEEE for approval as a standard and we don't expect much change in Draft 5.0. We plan to complete this recirculation ballot, clean up the draft and submit it to IEEE RevCom for final approval in time for their March 1993 quarterly review meeting. Based on this schedule, I would expect to see it approved and published by the IEEE mid-year 1993.

It is still my understanding that when P1224.2 and P1327.2 are approved by the IEEE, the US TAG to ISO/IEC JTC1 will propose that they be accepted by ISO as Draft International Standards (DIS) and balloted directly (fast tracked).

In Closing ...

There's quite a bit of work remaining, such as coordinating the recirculation and wrapping up loose ends for submission to IEEE RevCom. The group is not planning to meet in New Orleans in January.

Report on The Distributed Security Study Group

Dave Rogers <drogers@datlog.co.uk> reports on the October 19-23, 1992 meeting in Utrecht, NL

The POSIX Distributed Security Study Group (DSSG) met for the third and last time in Utrecht. This is the end of the six month lifetime of the study group. The group continued to be well supported and the Utrecht meeting brought a few new faces into the group, particularly European, but also a Canadian.

The DSSG made progress with the approach of defining a security framework for POSIX by mapping the ECMA "Open Systems Security – A Security Framework" onto a POSIX environment with encouraging results. The draft framework produced has been used to make an initial identification of the services requiring Application Program Interfaces and has mapped known existing or emerging implementations onto the APIs identified. Other standards activities in this area have also been identified.

A white paper titled "A Distributed Security Framework For POSIX" has been published presenting the work done to date with the specific objective of stimulating discussion and comment.

The DSSG has recommended the formation of a new POSIX working group to produce a "Guide to Security within Distributed POSIX Systems" using the white paper produced by the DSSG as the base document. A project authorization request (PAR) for this work has been submitted and will be considered by the POSIX SEC at the January meeting. An objective of this Guide is to produce a definition of the security services and APIs required throughout POSIX so that the adequacy of future PARs on meeting the defined security requirements can be assessed.

If anyone is interested in obtaining a copy of the white paper or wants more information then contact the DSSG Chair:

David Rogers
+44 81-863-0383 or
+44 256-59222 x4083
Data Logic Ltd
Queens House
Kymberely Road
Harrow,
Middx HA1 1YD
UK
email: drogers@datlog.co.uk

Report on IEEE Standards Board

Mary Lynne Nielsen <m.nielsen@ieee.org> reports on the September, 1992 meeting in New York, NY

September's meeting was unusually busy for TCOS, with lots of new project authorization requests (PARs) due to mirving activities and, at last, approval of one of the key components of POSIX. Decisions in the area of JTC1 funding will also have an impact on TCOS work.

[Ed. – TCOS is the technical committee within the IEEE responsible for developing the POSIX standards.]

At Long Last....

The IEEE Standards Board Review Committee (RevCom) approved P1003.2 and P1003.2a as IEEE standards at this meeting. POSIX.2 covers the shell and utilities for a POSIX system, while 1003.2a covers the user portability extensions for the shell. This pile of over 1300 pages of material is now in the publication process and should be available in the spring of 1993. Congratulations to all involved!

Note to any who actively work with this committee: as of the March 1993 meeting, only the new RevCom submittal forms will be accepted. Make sure you're using the form dated 9/92.

NesCom Actions Everywhere

The IEEE Standards Board New Standards Committee (NesCom) dealt with a whopping 14 Project Authorization Requests (PARs) from TCOS at this meeting. Twelve of these 14 came from the mirving, or splitting up, of three existing TCOS projects. Why did that have to happen? Well, mostly from a lot of resolutions either from various committees of ISO/IEC JTC1 (all of which basically translates to "the international group working on TCOS projects") or from TCOS itself. These resolutions say that it's better to have this work, have it all completed at the same time, and have it in bite-sized, somewhat digestible chunks, rather than receiving one huge document that takes a great deal of time to prepare. (For example, POSIX.2 was in ballot for three years).

What that means is that the various PARs in TCOS will often have to split into at least two and usually four parts: a base standard that is language independent, its test methods, a related language binding (usually C at first), and its test methods. While there is some debate as to the merits of this method, this practice is now being put into force in TCOS.

The first documents to be produced in this manner will be the 1224 series of standards (which is now the 1224, 1326, 1327, and 1328 standards). There is a strong indication that these standards will make the March 1993 RevCom meeting for approval, and the PARs for their mirving were approved at this meeting.

Also approved was a PAR for the revision of IEEE Std 1003.3-1991, the base standard on how to describe test methods for POSIX. Due to the expansion of testing to all TCOS (not just POSIX) standards and the need for test methods for new types of documents like profiles, the committee felt that it was time to start work on a revision of this standard.

In an attempt to control the bewildering expansion of 'dot' projects, a new numbering system will be employed for the POSIX testing standards. They will be numbered 2003.x, in parallel with the base standard they are testing. This revision is therefore numbered P2003.

Finally, P1003.19 was finally approved at this meeting, when NesCom at last received the reassurances they wanted that this work was not an infringement on the X3 work on the Fortran language itself. As such, the PAR for work on a Fortran 90 binding to POSIX.1 has at last gained clearance to go ahead.

Is It TransCom. . . or Isn't It?

TransCom, the IEEE Standards Board Transnational Committee, has voted to changed its name to IntCom, the IEEE Standards Board International Committee, an action that was also approved by the Board. It seems that the term "transnational," while used in the IEEE bylaws to define the scope of the IEEE, is very confusing to the members of this committee and to the people they speak to about their work. (My understanding is that the term means "without borders.") They feel that the word "international" far better suits the activities they undertake, which is to coordinate IEEE standards activities with non-US standards organizations.

In addition, Trans/IntCom continued to work on a guide for synchronizing work with ISO/IEC JTC1, a plan that recognizes the methods used by POSIX to move its standards forward in this arena. This guide should hopefully be approved in December.

IT Funding

As mentioned in earlier snitch reports, the Standards Board has been wrestling with an action

from ANSI that proposes having the groups involved in JTC1 activities support the secretariats of JTC1 that ANSI maintains. The IEEE Standards Board, representing one of the major groups involved, created an ad hoc committee to explore resolutions to this issue. TCOS supplied information to this committee in the form of a resolution expressing their position, while the committee examined the financial and legal aspects of this question. They also examined if this funding conflicted with the expressed goals of the IEEE Standards Board Strategic Objectives.

The committee submitted its final report at this Board meeting. In it, they felt that these funds could be collected without any negative impact on the legal aspects, financial aspects, or stated objectives of the IEEE Standards Board. The report recommended that IEEE staff work with the standards committees in designing and implementing procedures for the collection and administration of participation fees assessed to IEEE participants for these secretariats. The report also stated that each standards committee should decide on its own procedures for fund collection, but they should be encouraged very strongly to standardize on one or two methods for collecting fees.

One note on this: TCOS discussed this situation at its October meeting in Utrecht, and the following methods for collecting funds were approved by the TCOS Standards Executive Committee (SEC): an increase in balloting fees; an increase in NAPS mailing costs; a reduction in meeting services (such as Friday lunches); and a fee imposition for meetings held independently of the regular TCOS meetings. It was felt that this system would distribute the burden of raising these funds equitably among those who attend meetings and those who do not but who participate in the process through mailings.

Awards and Recognition

Three TCOS members received awards from the IEEE Standards Board, called IEEE Standards Medallions, in recognition of their contributions to standards development. They are Donn Terry, the former chair of POSIX.1, Hal Jespersen, the chair of POSIX.2, and Roger Martin, the chair of the TCOS Steering Committee on Conformance Testing (SCCT) and the former chair of the POSIX.3 (Test Methods) working group. Congratulations to them all.

NesCom Approvals

New PARs

P1003.19 Standard for Information Technology – POSIX Fortran 90 Language Interfaces – Part 1: Binding for System Application Program Interface (API)

P2003 Standard for Information Technology – Test Methods for Measuring Conformance to POSIX

Revised PARs

P1224 Standard for Information Technology – Open Systems Interconnection (OSI) Abstract Data Manipulation – Application Program Interface (API) [Language Independent]

P1224.1 Standard for Information Technology – X.400 Based Electronic Messaging Application Program Interfaces (APIs) [Language Independent]

P1224.2 Standard for Information Technology – Directory Services Application Program Interface (API) – Language Independent Specification

P1326 Standard for Information Technology – Test Methods for Measuring Conformance to Open Systems Interconnection (OSI) Abstract Data Manipulation – Application Program Interface (API) [Language Independent]

P1326.1 Standard for Information Technology – Test Methods for Measuring Conformance to X.400 Based Electronic Messaging Application Program Interface (API) [Language Independent]

P1326.2 Standard for Information Technology – Test Methods for Directory Services Application Program Interface (API) – Language Independent Specification

P1327 Standard for Information Technology – Open Systems Interconnection (OSI) Abstract Data Manipulation C Language Interfaces – Binding for Application Program Interface (API)

P1327.1 Standard for Information Technology – X.400 Based Electronic Messaging C Language Interfaces-Binding for Application Program Interface (API)

P1327.2 Standard for Information Technology – Directory Services Application Program Interface (API) – C Language Specification

P1328 Standard for Information Technology – Test Methods for Measuring Conformance to Open Systems Interconnection (OSI) Abstract

Data Manipulation C Language Interfaces – Binding for Application Program Interface (API)

P1328.1 Standard for Information Technology – Test Methods for Measuring Conformance to X.400 Based Electronic Messaging C Language Interfaces – Binding for Application Program Interface (API)

P1328.2 Standard for Information Technology – Test Methods for Directory Services Application Program Interface (API) – C Language Specification

RevCom Approvals

P1003.2 Standard for Information Technology – Portable Operating System Interface (POSIX) – Part 2: Shell and Utilities

P1003.2a Standard for Information Technology – Portable Operating System Interface (POSIX) – Part 2: Shell and Utilities, User Portability Extension

Report on ISO WG15 (POSIX) Rapporteur Group on Co-ordination of Profile Activities

Kevin Lewis <klewis@gucci.enet.dec.com> reports on the October 23-24, 1992 meeting in Utrecht, NL

The IEEE Technical Committee on Operating Systems – Standards Subcommittee (TCOS-SS) forwards POSIX documents through an ANSI technical advisory group to ISO Working Group 15 (WG15) for approval as international standards. WG15 has a number of rapporteur groups, which are small groups of experts on various ISO POSIX related topics.

This was the third meeting of the Rapporteur Group on Coordination of Profile Activities (RGCPA). It was my first. The meeting lasted a day and a half. There were actually more observers in the room than members. About 15-18 people attended, of which 75% were IEEE POSIX attendees. Seeing all the familiar faces from a week of IEEE POSIX meetings underscored the high percentage of overlap between the IEEE and ISO POSIX working groups.

The work of this Rapporteur group is to co-ordinate profiling activities that would be of interest to WG15 as follows:

- the process of addressing user requirements for profile harmonization,
- the development of the appropriate approach to sub-setting WG15 standards within profiles,

- the treatment within profiles of the options that exist within a standard that is part of the profile.

Recognizing that there are other organizations dealing with profile issues, the group put forward a resolution to WG15 that the TCOS Profile Steering Committee and X/Open be encouraged to establish Category C liaison with WG15 RGCPA.

Reflecting back on this meeting, it seemed to me that the real purpose of this group is to serve as a radar, seeking out any and all profile activities anywhere in the globe that would be pertinent to the work of WG15 and SC22. From my own vantage point, it appeared to be accomplishing his purpose.

The next meeting of this group will be on May 10-11, 1993 in Heidelberg, Germany.

The Elusive JTC1

John Hill <hill@prc.unisys.com>

Quite often in reading articles concerning standards for information technology the term "JTC1" is encountered. This article defines the term, describes its activities, and puts JTC1 in context.

Until late 1988 there were multiple confusing processes for developing worldwide standards for information technology. Some standards, such as those for equipment and electrotechnical matters, were developed by the IEC. IEC is the acronym for the French equivalent of the "International Electro-technical Commission." Other standards, such as those for media and programming languages, were developed under the auspices of ISO. ISO is the commonly used name for the French "international standards organization."

The source of the confusion about ISO and IEC was largely at the detailed level of standards development, and stemmed from the fact that there was overlap of the work of the two organizations.

In the middle 1980s, thanks largely to the efforts of Ed Lohse, late of Burroughs Corporation, activities to rationalize the situation were started in earnest. The product of these activities is the ISO/IEC Joint Technical Committee 1, or JTC1.

JTC1 is the first, and currently the only, technical committee that is jointly managed by ISO and IEC. Devising the scheme for joint management of JTC1 was a formidable task. Here were two organizations whose generalized aims were similar

but operated in dissimilar fashions in key procedural areas.

The situation was sufficiently complex that they decided that separate procedures for JTC1 were to be developed and approved. This document is known as the JTC1 Directives. (The JTC1 Directives can be obtained from the JTC1 secretariat, ANSI.)

So much for the framework. Now for the current organization and program of work of JTC1 and its subgroups.

First, you must understand that the members of JTC1 are referred to as member bodies. There are two types of member bodies: national bodies, and liaisons. There are 42 national member bodies. (24 are primary, and 18 are observer). As an example, the USA, as represented by ANSI, is a national body member of JTC1. There are others, including France (AFNOR), Germany (DIN), and Sweden (SII).

The matter of liaison members is a bit more complicated. There are 14 internal liaisons. These are subgroups of ISO or IEC that have interest in the work of JTC1. There are also 19 external liaisons. ECMA, the European Computer Manufacturers Association, is a representative example of a liaison member of JTC1. One interesting sidelight to this is that most nations have some sort of umbrella-like standards organization that can be designated as the country's representative in JTC1. These national umbrella standards organizations operate within their own countries according to their own rules and procedures. JTC1, while insulated from member countries' internal operations, is nonetheless aware of them.

So the membership of JTC1 is either national (i.e., by country) or notified liaison. There is no concept of "organizational" or corporate membership. Similarly, there are no individual members. Many national bodies operate internally on the basis of organizational membership. Some operate on the basis of individual membership. The umbrella organization in the USA, ANSI, accredits organizations and committees to develop standards for it. Membership in some of these is organizational, such as X3. In some it is individual, such as the IEEE, the Institute of Electrical and Electronic Engineers.

For the most part the work of JTC1 itself is managerial in nature. JTC1 focuses on matters like:

- project initiation,
- subgroup establishment (and disposal),
- document approval.

The technical work of JTC1 is really accomplished by its subgroups. Broadly speaking, there are three types of JTC1 subgroup. These are special working groups (SWG), study groups (SG), and subcommittees (SC).

SWGs are typically established to perform some specific task and are often non-technical in nature. Examples include the SWG-P that deals with JTC1 procedures, and SWG-FS that deals with functional standards (often called international standardized profiles, or ISPs). *[Ed.— SWG-FS is sometimes referred to simply as SGFS.]* SWG-FS has developed Technical Report (TR) 10000 that describes procedures for the development of open systems interconnection (OSI) ISPs. SWG-FS is currently revising TR10000 in order that it incorporate procedures for managing the development of ISPs for open systems environments.

There have been two special study groups established by JTC1. Each was given a specific charter and assigned specific deliverables. Neither exists today since they completed their assignments. The two study groups were MSG-1 (management study group) and TSG-1 (technical study group). TSG-1 focused on interfaces for application portability.

The most enduring subgroup type is the subcommittee (SC). SCs tend to be organized around functional topics. An SC's typically focuses on a single technical subject area. The detailed standards development work of an SC takes place within the working groups (WG) within an SC.

One way to better grasp the activities of JTC1 is to group the SCs. There are four convenient groupings:

- application elements,
- systems,
- equipment and media,
- systems support.

A complete list of these SCs follows the article, grouped according to the above list.

The scope of JTC1 is extensive. Virtually all standards used in modern information technology systems receive their worldwide endorsement by JTC1. This has simply been an overview. There are a multitude of detailed projects that collectively specify the full depth of the technical program of JTC1.

ISO/IEC JTC1 Subcommittees:

Application Elements

SC1 (Vocabulary): To collect and coordinate the usage of terminology by all groups within JTC1. The Dictionary Group!

SC7 (Software Engineering): To define standardized tools to development software.

SC14 (Representation of Data Elements): To codify data elements such that their common definitions can be used to exchange data.

SC22 (Languages and Application Environments): Programming Language and Operating Environment standards.

Systems

SC6 (Telecommunications and Information Exchange): Standards for telecommunications and OSI, (systems functions, procedures and parameters, as well as the conditions for their use) for the four OSI layers that support the transport service. Done in effective cooperation with CCITT.

SC18 (Text and Office Systems): Standardization of functionality that simplifies text editing and other office related subjects.

SC21 (OSI Information Retrieval, Transfer and Management): Development of standards for the upper layers of the Open Systems Interconnection (OSI) model. Also included are database management systems, information resource management systems (IRDS), and open distributed processing standards (ODP).

SC26 (Microprocessor Systems): Development of standards used in microprocessor systems including basic hardware, bus and allied interfaces.

Equipment and Media

SC11 (Flexible Magnetic Media for Digital Data Interchange): Development of standards for diskettes and cartridges. The unrecorded (raw media) as well as the recording standards are both included.

SC15 (Labeling and File Structure): Standardization of file allocation and directory information used for all types of recorded media.

SC17 (Identification Cards and Related Devices): Standards for cards such as credit

and debit cards including the physical, electrical, and magnetic properties. Intelligent (IC) cards are also covered.

SC23 (Optical Digital Data Disks): Development of optical media standards including the unrecorded (raw) media as well as the recording onto and reading from those media. Both write once (WORM) and rewritable media are included.

SC28 (Office Equipment): Standardization of equipment commonly used in office settings. This includes printers and the quality of their output.

Systems Support

SC2 (Character Sets and Information Coding): Standards for the bit and byte coded representation of elements of various identified types of information, for interchange mainly at the application level, i.e., all aspects of sets of graphic and control characters.

sC27 (Security Techniques): Development of standards for security, such as encryption and verification.

SC29 (Coded Representation of Picture, Audio and Multimedia/Hypermedia Information): Standardization of complex (i.e., more difficult than characters) data representation. Data compression without the loss of information is also handled here.

The Bookworm

by Peter H. Salus
Sun User Group, Inc.
<peter@usenix.org>

Slicing a layer cake

My guess is that no one who reads what I write has any doubt about my attitude towards OSI: I don't like it. However, it is also clear that with so many international PTTs supporting this layering, it will win—though I think it can only be effective by sitting under TCP/IP [or a successor suite of protocols].

Having said this, I'd like to turn to *The Basics Book of OSI and Network Management*. This is one of a series of small pamphlets published by Addison-Wesley for the "Motorola University Press." The latter should not be confused with MIT Press or Oxford University Press. The blurb on the rear cover of this 59-pager tells me that it is "designed for MIS people, beginners and experienced alike, who want a quick but accurate summary of what network management is..."

Well, this is a book that can be read quickly. It is written in a breezy fashion. And it contains such statements as: "OSI Management is a software application of the OSI application layer. Its mission is to manage the *managed objects* ..." (p. 25). I may be nasty, but I'm always willing to let a book hang itself. This fatuous pamphlet ends with: "And because network management is, at heart, a very logical activity, commonsense questions will take you the rest of the way" (p. 57).

Well, commonsense led me to look for DNS and domain and name in the index. Neither is there.

Domain Name System

My questions next led me to Albitz and Liu's *DNS and BIND* from O'Reilly and Associates. While many folks have criticisms of DNS and even more feel that BIND is another typical undergraduate project from Berkeley, they make up basic building blocks for the Internet. Basic enough that I would have expected a book on "Network Management" to contain them. *DNS and BIND* explains how foo.com gets translated into 123.45.67.89. Furthermore, Albitz and Liu have limned the contents in such a manner that (a) you can go through the chapters progressively or (b) look up something specific in their index.

Together with the recent TCP/IP volume (Craig Hunt) and Ed Krol's *Whole Internet...*, O'Reilly now offers a range of documentation to mail and Internetting that most readers will find irresistible.

Remember Calculus?

When Nakamura's *Applied Numerical Methods in C* clunked onto my desk, I cringed. Did I even want to think about Lagrange and Newton; polynomials and nonlinear equations; eigenvalues and parabolic partial differentials after all these years? And, anyway, who cares if the guys who think about that stuff switch from Fortran to C?

So I began reading.

Nakamura has done a really good job. This is a large, detailed, high-level work which should be genuinely useful for those involved in numerical analysis. It is intended as an advanced undergraduate/graduate textbook. There are occasional editing slips, but by and large this is a readable, comprehensible volume. Nakamura states that the programs "were tested with Microsoft QuickC," which I don't know. But they look as though they'd compile on any C compiler, though I think folks ought to be careful about those ANSI C headers.

Other Applications

Adventures in UNIX Network Applications Programming isn't much of an adventure. But I think that Rieken and Weiman have managed to write a volume that the applications programmer with little experience in network programming will find useful. However, I can see it as even more useful in an advanced undergraduate class or a beginning graduate-level one. Some nice aspects: Each chapter is labelled as to who wrote it and begins with an introduction and ends with a summary or a sample program.

Print Service

Sally A. Browning has done a respectable job of rewriting the materials in the SVR4 System Administrator's Guide and the SVR4 Guide for Intel Processors as well as various man pages to come up with *UNIX System V Print Service Administration*, which Prentice Hall has sent me in pre-publication form. I guess it will be out by the time you read this. It may even be at the San Diego USENIX. Anyway, this will be a handy addition to the library of those who can't recall what *lpstat* does or how to configure a remote printer. Or, more importantly, those whose managers have suddenly decided to "go UNIX," after years of other systems.

Updates and Revisions

Well, O'Reilly has been busy recently. In addition to the various new volumes I've mentioned, they've also revised, expanded, and updated several of their earlier volumes. Two that I've looked at and liked are *lex & yacc* and *MH & xmh*.

lex & yacc has acquired a third author (John R. Levine), who has made major changes where the volume of Tony Mason and Doug Brown of two years ago is concerned. His work appears to have been directed towards two goals: help for folks learning *lex*, and explanations for seasoned programmers attempting to understand what Lesk (and Eric Schmidt) and Johnson wrought. Now that most people (I think) use *flex*, it's also nice to be able to read what Levine says about it. This is more than just a mere update.

MH & xmh is also a revision of a first edition. There are lots of things that Jerry Peek has altered and improved. Unfortunately, the book is getting almost too big to handle comfortably. I appreciate the 45 pages on *mhook*, but wonder whether there might not have been 10. The purple pages at the back – *xmh* reference guide – are just great, though. I also liked the notes by Bruce Borden and Stockton Gaines on the early history of *MH*.

Writing intelligibly

Addison-Wesley has brought out a paperback version of *The Economist Style Guide*. As this is my all-time favo(u)rite – better by far than the *New York Times* or *Associated Press* guides, I want to personally thank A-W for making this available: every prospective author of materials for *login: and Computing Systems* should glance at this before submitting anything. There is too much jargon and flabby writing in the universe; we can do our bit to restrain this. *The Economist Style Guide* is also full of pithy examples:

Anticipate does not mean expect. Jack and Jill expected to marry; if they anticipated marriage, only Jill might find herself expectant. (p. 14)

The Basics Book of OSI and Network Management. Addison-Wesley, 1993. ISBN 0-201-56371-1. 59pp. \$9.75.

Shoichiro Nakamura, *Applied Numerical Methods in C*. Prentice-Hall, 1993. ISBN 0-13-042052-2. 604pp. .

Paul Albitz & Cricket Liu, *DNS and BIND*. O'Reilly & Associates, 1992. ISBN 1-56592-010-4. 381pp. \$29.95.

Bill Rieken & Lyle Weiman, *Adventures in UNIX Network Applications Programming*. John Wiley & Sons, 1992. ISBN 0-471-52858-7. 448pp. .

Sally A. Browning, *UNIX System V Print Service Administration*. Prentice Hall. 195pp.

John R. Levine, Tony Mason & Doug Brown, *lex & yacc*. O'Reilly & Associates, 1992. ISBN 1-56592-000-7. 366pp. \$29.95.

Jerry Peek, *MH & xmh*. O'Reilly & Associates, 1992. ISBN 1-56592-027-9. 728pp. \$29.95.

The Economist Style Guide. Addison-Wesley, 1992. ISBN 0-201-63200-4. 147pp. \$9.95.

USENIX SUMMER 1993 TECHNICAL CONFERENCE

JUNE 21-25, 1993
CINCINNATI
CONVENTION CENTER
CINCINNATI,
OHIO

IMPORTANT DEADLINES

- **Registration Materials Available:**
End-March 1993
- **Pre-Registration Savings Deadline:**
Monday, June 1, 1993
- **Hotel Savings Deadline:**
Monday, June 1, 1993

CONFERENCE SCHEDULE

- **Tutorials:**
Mon. and Tues., June 21-22, 1993
- **Technical Sessions:**
Wed. through Fri., June 23-25, 1993
- **Birds-of-a-Feather Sessions:**
Tues. through Thurs.,
June 22-24, 1993
- **USENIX Reception:**
Thursday evening, June 24, 1993
- **Vendor Display:**
Wed. and Thurs., June 23-24,
12:00 pm-6:00 pm



ANNOUNCEMENT/CALL FOR PARTICIPATION

TUTORIAL PROGRAM

Monday and Tuesday, June 21-22, 1993

The USENIX Association's well-respected program offers you introductory and advanced, intensive yet practical tutorials. Courses are presented by skilled teachers who are hands-on experts in their topic areas. At Cincinnati USENIX will offer tutorials such as: Topics in systems administration, Distributed computing: DCE, DME, DFS, UNIX programming tools, Systems and network security, Kernel internals: OSF/1, SVR4, 4.4BSD, Developing and debugging X-based applications, Network program maintenance and design, Introductions to Perl and systems programming, Microkernel technologies, Overview of GUI technologies and builders

TECHNICAL SESSIONS

Wednesday-Friday, June 23-25, 1993

"Evolving New User Interface Technologies for UNIX" is the theme of the Summer 1993 Technical Sessions. As always at USENIX, we will explore new and interesting developments in open operating systems. But in Cincinnati we're particularly examining the evolution of operating system capabilities to support new and effective user interfaces. Communicating with the user is a real-time problem, why aren't we using the emerging real-time capabilities of UNIX to support it? Are UNIX byte-string files adequate, or do we need a generalized file attribute model? Can users really navigate a file name space that is a rooted tree of all the files in the Internet? Radical thinking is needed.

KEYNOTE SPEAKER

Bruce Tognazzini, SunSoft, Inc.

Our keynote speaker has been a long-time customer of the operating system support for the user interface. He has been designing man-machine interfaces for better than 30 years. He spent the last 14 years at Apple where he led at various times both the Apple II and Macintosh human interface efforts before moving to SunSoft earlier this year.

INVITED TALKS

Invited talks provide introductory and advanced information about a variety of interesting topics, such as using standard UNIX tools, tackling system administration difficulties, or employing specialized applications. Suggestions and proposals are welcomed by the Invited Talks Coordinators: Tom Cargill, Consultant (303) 494-3239; Bob Gray, US WEST (303) 541-6014 (or e-mail to: ITusenix@usenix.org).

BIRDS-OF-A-FEATHER SESSIONS

Schedule a BOF in advance or on-site; contact the USENIX Conference Office, (714) 588-8649, e-mail: conference@usenix.org

WORKS-IN-PROGRESS REPORTS

Present your newly completed, on-going or not fully developed work in a 10-minute WIP. Gain valuable discussion and feedback. Schedule your report in advance via e-mail to wip@usenix.org or on-site with WIP Coordinator, Peg Schafer, Bolt Beranek & Newman, Inc.

SUMMER 1993 VENDOR DISPLAYS

The Vendor Display will provide a relaxed environment in which conference attendees and technically savvy vendor representatives have time to talk together and learn from one another. This is an exceptional opportunity for

(continued on reverse side)

PROGRAM COMMITTEE

► Program Chair:

David S.H. Rosenthal,
SunSoft Inc.

Matthew Blaze, *AT&T Bell Laboratories*

Nathaniel Borenstein, *Bellcore*

Bob Gray, *US WEST Advanced Technologies*

Steve Kleiman, *SunSoft Inc.*

M. Kirk McKusick, *University of California, Berkeley*

Jeffrey Mogul, *Digital Equipment Corporation*

J. R. Oldroyd, *Instruction Set*

Pat Parseghian, *AT&T Bell Laboratories*

Dennis Ritchie, *AT&T Bell Laboratories*

DATES FOR REFEREED PAPER SUBMISSIONS

- Extended Abstracts Due:
February 2, 1993
- Notification to Authors:
February 27, 1993
- Camera-Ready Papers Due:
April 14, 1993

FOR REGISTRATION INFORMATION

Materials containing the technical and tutorial program, conference registration, hotel and airline discount and reservation information will be mailed at the end of March 1993. If you wish to receive the registration materials, please contact:

- **USENIX Conference Office**
22672 Lambert St., Suite 613
El Toro, CA 92630 U.S.A.
(714) 588-8649,
FAX (714) 588-9706
E-mail: conference@usenix.org

USENIX

receiving feedback on new development from USENIX's technically astute conference attendees. If your company would like to display its products and services, please contact Cynthia Deno (408) 335-9445, FAX (408) 335-2163, e-mail: cynthia@usenix.org

REFEREED PAPER SUBMISSIONS

Authors of papers to be presented at the technical sessions and published in the proceedings must submit one copy of an extended abstract via at least two of the following methods:

- E-mail to summer93papers@usenix.org
- FAX to (510) 548-5738
- Mail to: Summer 93 USENIX, USENIX Association, 2560 Ninth St., Suite 215, Berkeley, CA 94710 U.S.A.

The object of an extended abstract is to convince the reviewers that a good paper and 25-minute presentation will result. Reviewers don't have time to read full papers. They need to know that authors:

- are attacking a significant problem.
- are familiar with the current literature about the problem.
- have devised an original solution.
- have implemented it and, if appropriate characterized its performance.
- have drawn appropriate conclusions about what they have learned and why it is important.

As at all USENIX conferences, papers that analyze problem areas and draw important conclusions from practical experience are welcome. Note that the USENIX conference, like most conferences and journals, considers it unethical to submit the same paper simultaneously to more than one conference or publication or to submit a paper that has been or will be published elsewhere.

The extended abstract must be 5 manuscript pages (single side) or less in length. Only the first 5 pages of your submission will be sent to the reviewers. The full paper may be attached to the extended abstract; it will not be sent to the reviewers but may be helpful during final evaluation.

The extended abstract should represent the paper in "short form." It should include the abstract as it will appear in the final paper. Supporting material may be in note form. Authors should include references to establish that they are familiar with the literature, and, if appropriate, performance data to establish that they have a working implementation and measurement tools.

Every submission should include one additional page containing:

- The name, surface mail address, daytime and evening telephone numbers, e-mail address and (if available) fax number of one of the authors, who will act as the contact to the program committee.
- An indication of which, if any, of the authors are full-time students.
- A list of audio/visual equipment desired beyond a microphone and an overhead projector.

Enquiries about submissions to the USENIX Summer 1993 Conference may be made by e-mail to david@usenix.org or to (510) 528-8649. You may request a sample extended abstract by telephoning (510) 528-8649 or by fax to (510) 548-5738.

USENIX, the UNIX and Advanced Computing Systems Professional and Technical Association.

3rd USENIX MACH SYMPOSIUM

APRIL 19-21, 1993
EL DORADO HOTEL,
SANTA FE,
NEW MEXICO

IMPORTANT DATES

- **Registration Materials Available:**
End-January, 1993
- **Tutorial Program:**
Mon., April 19, 1993
- **Technical Sessions:**
Tues., April 20 - Wed., April 21, 1993
- **Birds-Of-a-Feather Sessions:**
Tues., April 20, 1993
- **USENIX Reception:**
Tues. evening, April 20, 1993



The use and influence of Mach on the operating systems community continues to grow. From its beginnings as a small research project, Mach has spread to become the basis for commercial products from a variety of vendors and a key component of innovative research efforts in both academic and industrial environments. At the same time, research and development continue to evolve Mach itself. The community of researchers and developers working with Mach is proving to be a very productive source of innovative systems.

Activity in this field has been sufficiently wide-spread that the USENIX Association is pleased to once again sponsor a Mach symposium to bring together researchers, engineers, vendors and users of Mach systems. We encourage discussion of all past and present Mach-related research, development, production and applications activities.

SYMPOSIUM OVERVIEW

The first day of the three-day Mach Symposium will be devoted to tutorials. The following two days will concentrate on presentations of refereed papers on past and present Mach-related work. Long breaks between presentations will provide opportunities for informal discussion. Panel discussions will be offered, and some time made available for reports of work-in-progress. Birds-of-a-Feather sessions will be available in the evenings. (You may schedule a Birds of a Feather session by contacting the USENIX Conference Office.)

TUTORIAL PROGRAM

The tutorials at the USENIX 1993 Mach Symposium are geared predominantly toward programmers and systems programmers interested in learning more about the Mach Kernel and in using Mach. These intensive classes are given by excellent presenters who are also experts in their field.

The four tutorials are each a half-day long. Attendees may choose one morning tutorial and one afternoon tutorial, for a total of two. USENIX tutorials typically sell out, and on-site registration is available only if space is available. Attendees are strongly encouraged to pre-register for these tutorials.

➤ AM1: Introduction to the Mach 3.0 Microkernel

Instructor: Tom Doeppner, *Brown University*

This tutorial will provide a quick introduction to the internals of Mach 3.0. Although the tutorial is oriented to those with some exposure to earlier versions of Mach, it can be profitably taken by those who have had no prior exposure to Mach.

We will discuss the basic structure of Mach and go over Mach's fundamental abstractions, including tasks, threads, ports, messages and memory objects. We will examine two important optimizations, lazy evaluation and continuations, and discuss how they speed up virtual copy optimizations and intra-machine remote procedure calls. Finally, we will look at how traditional operating systems, such as UNIX, can be (and are) implemented in user mode on top of Mach.

Thomas W. Doeppner, Jr. received his Ph.D. in Computer Science from Princeton University. He has been on the faculty at Brown University since 1976. His research interests are in operating systems and parallel programming. He has lectured extensively on UNIX internals over the past seven years for the Institute for Advanced Professional Studies and he authored the multi-day OSF/1 internals course offered by OSF.

➤ AM2: Mach IPC and Mig Programming

Instructor: Richard Draves, *Microsoft Corporation*

This tutorial is intended for programmers interested in using Mach IPC or constructing client-server systems with Mach. No prior knowledge of Mach is necessary. Prior experience with client-server programming is helpful but not required.

The Mach 3.0 IPC facility provides message-oriented, capability-based inter-process communication. The interface supports several different styles of communication, including remote procedure call, object-oriented distributed programming, and asynchronous messages. Mach IPC is transparent and extensible. The Mig stub generator produces code to pack and unpack messages, given a high-level definition of a message interface.

This tutorial provides an introduction to Mach 3.0 IPC and Mig. It first covers the concepts underlying Mach IPC and then teaches the use of Mach IPC via Mig. Emphasis is on the construction of client-server systems.

(continued on reverse side)

PROGRAM COMMITTEE

Program Chair:

David Black,
Open Software Foundation

David Golub, Carnegie Mellon University

Alan Langerman, Orca Systems, Inc.

Jay Lepreau, University of Utah

Avadis Tevanian, Jr., NeXT, Inc.

TO SUBMIT PROGRAM PROPOSALS

You may submit an invited talk or panel proposal, arrange participation in a panel, and/or schedule a Work in Progress Report by contacting the program chair:

David Black
Research Institute
Open Software Foundation
1 Cambridge Center, 11th Floor
Cambridge, MA 02142 U.S.A.
(617) 621-7347
FAX: (617) 621-8696
E-mail: dlb@osf.org

FOR REGISTRATION INFORMATION

Materials containing all details of the technical and tutorial program, conference registration, hotel and airline discount and reservation information will be mailed at the end of January of 1993. If you wish to receive the registration materials, please contact:

USENIX Conference Office
22672 Lambert St., Suite 613
El Toro, CA 92630 U.S.A.
(714) 588-8649
FAX: (714) 588-9706
E-mail: conference@usenix.org

USENIX

The UNIX and Advanced Computing Systems Professional and Technical Association.

Topics covered include: ports, port rights, messages, port sets, out-of-line memory, Mig type definitions, routine, simple-routine, netname interface, client-server construction.

Richard Draves, as a graduate student at Carnegie Mellon University, designed and implemented the Mach 3.0 kernel IPC mechanism. He also implemented the current version of Mig. Richard Draves is now a Researcher at Microsoft Corporation.

PM1: Porting and Modifying the Mach 3.0 Microkernel

Instructor: Bob Wheeler, Carnegie Mellon University

This tutorial is aimed at programmers and system programmers familiar with basic operating system design and concepts. Prerequisites include a working knowledge of C and a familiarity with some form of assembly language.

This tutorial will be a practical introduction to Mach 3.0 internals for people wanting to port the microkernel to a new architecture or work with an existing implementation. Coverage will be of the various machine-dependent routines in Mach with an emphasis on how these fit into and are used by the rest of the kernel and server. The goal of the tutorial is to have participants leave with enough knowledge to start working with the kernel and server sources and for participants to understand the machine-dependent routines and how they fit into the overall design of the system.

Topics to be discussed will include:

- ◆ The layout of the kernel and server sources
- ◆ The Mach build tools
- ◆ MachUs virtual memory system
- ◆ The pmap module
- ◆ Saving and restoring kernel and user state
- ◆ System call emulation
- ◆ Mach devices
- ◆ The emulation library
- ◆ The single server
- ◆ Debugging

Bob Wheeler is a graduate student at Carnegie Mellon University working on the Mach project. His most recent work has been on changing MachUs virtual memory system to accommodate virtually indexed caches. Before moving to CMU he worked at the University of Utah on the Mach 3.0 port to the Hewlett Packard PA-RISC architecture. He has worked on kernels for embedded systems, BSD and Mach. He holds degrees in both Mathematics and Physics from the University of Utah.

PM2: Mach External Memory Managers: Principles and Practice

Instructor: David L. Black, Open Software Foundation

This tutorial assumes some familiarity with Mach 3.0 IPC and Mig. Emphasis is placed on the role and responsibilities of an external memory manager in the Mach system. The tutorial teaches the student how to write an external memory manager. It includes an example of an external memory manager.

The following topics will be covered:

- ◆ The Mach external memory management architecture and interface
- ◆ Writing an external memory manager
- ◆ Using advanced memory management features
- ◆ Applications of external memory management
- ◆ The default memory manager
- ◆ Performance and robustness techniques

David L. Black is a Research Fellow at the Open Software Foundation's Research Institute in Cambridge, MA. He received his doctorate in Computer Science in 1990 from Carnegie Mellon University, where he was one of the key designers and implementors of Mach. At OSF, he continues to work on Mach (including improvements to external memory management) in cooperation with the Mach project at CMU. Dr. Black also holds an MS in Computer Science from CMU and an MA in Mathematics from the University of Pennsylvania.

SYMPOSIUM TOPICS

Topics to be explored may include, but are not limited to:

- ◆ Applications and support for programming languages
- ◆ Mach 2.5 and related systems (e.g., OSF/1)
- ◆ Mach 3.0 and servers
- ◆ Mach-based operating system implementation and emulation
- ◆ Use of Mach subsystems in other operating systems
- ◆ Multiprocessor and parallelization experiences
- ◆ Distributed systems, including multi-computers, clusters, etc.
- ◆ Real time
- ◆ Security
- ◆ Performance
- ◆ Productization experiences
- ◆ Comparisons of Mach with other operating systems, e.g., Chorus, Sprite, Amoeba, V, and, of course, UNIX
- ◆ Future work

2nd USENIX SYMPOSIUM ON MICROKERNELS & OTHER KERNEL ARCHITECTURES

SEPTEMBER 20-22, 1993
HILTON BEACH &
TENNIS RESORT
SAN DIEGO, CALIFORNIA

DATES TO REMEMBER

Dates for Refereed Paper Submissions

- Extended abstracts due:
April 26, 1993
- Notification to authors:
May 26, 1993
- Camera-ready final papers due:
July 8, 1993

Registration Materials Available:

- June, 1993



ANNOUNCEMENT/CALL FOR PAPERS & PARTICIPATION

Proponents of microkernels claim that the use of this kind of technology is the inevitable next step in the engineering of operating systems. Their claim is microkernels bring the ability to support new hardware architectures and applications with no loss of performance. Whether or not this is true, this type of operating system architecture is being increasingly adopted by both industry and research.

Following the success of last year's Symposium, USENIX is pleased to announce the second USENIX Symposium on Microkernels and Other Kernel Architectures. This Symposium is aimed at exploring the different approaches to microkernels and the tradeoffs and benefits associated with each. Of particular interest is the question of whether microkernel architecture does lend itself to the support of new kinds of applications or operating systems which would be difficult or impossible to support under another operating system model.

TUTORIALS

September 20, 1993

The first day of this Symposium will feature a two track tutorial program. Expert-led tutorials will cover topics, such as current and forthcoming microkernels, of interest to the microkernels community.

TECHNICAL SESSIONS

September 21-22, 1993

The next two days will be devoted to presentation of papers from the industrial and research communities. These papers will be formally reviewed and selected by the Program Committee. The papers will be published in the Proceedings, distributed free to technical session attendees and available for purchase after the symposium from the USENIX Association.

SYMPOSIUM TOPICS

Papers are being solicited on microkernels, kernel architectures and what these bring to particular applications. Both positive and negative experiences are welcome. Topics include, but are not limited to:

- Performance and Optimization
- Fault Tolerance and High Availability
- Real-Time on Microkernels
- Scalability
- Distribution
- Evolution of Kernel Architecture
- Positive and Negative Experiences
- Use of Microkernels to Support Non-Traditional Applications
- Embedded or Dedicated Applications
- Applications Supported Directly by Microkernel

SUBMISSIONS

If you are interested in submitting a paper for the technical sessions, please submit an extended abstract. The object of an extended abstract is to convince the Committee that a good paper and 25-minute presentation will result.

They need to know that the authors:

- are attacking a significant problem.
- are familiar with the current literature about the problem.
- have devised an original solution.

(continued on reverse side)

PROGRAM COMMITTEE

➤ Program Chair:

Lori S. Grob
Chorus systèmes
grob@chorus.fr/grob@usenix.org
Brian Bershad
Carnegie Mellon University
bershad@cs.cmu.edu
Michael L. Powell
Sun Microsystems Laboratories
michael.powell@eng.sun.com

REFEREED PAPER SUBMISSION DATES

- Extended abstracts due:
April 26, 1993
- Notification to authors:
May 26, 1993
- Camera-ready final papers due:
July 8, 1993



The UNIX and Advanced
Computing Systems Professional
and Technical Association.

- have implemented it and, if appropriate, characterized its performance.
- have drawn appropriate conclusions about what they have learned and why it is important.

The extended abstract should include the abstract as it will appear in the final paper, and represent the paper in "short form." Supporting material may be in note or outline form. Authors should include references. It should be clear from the abstract whether the paper represents a design, an implementation or a system that is in wide use.

Note that the Program Committee considers it unethical to submit the same paper simultaneously to more than one conference or publication, or to submit a paper that has been or will be published elsewhere without including that information with the submission.

HOW TO SUBMIT

Your submission of one copy of an extended abstract will be accepted by fax, mail, or e-mail. E-mail is greatly preferred.

- E-mail to grob@usenix.org
- Fax to +33 1 30 57 00 66
- Mail to:
Microkernels
USENIX Association
2560 Ninth St., Suite 215
Berkeley, CA USA 94710

The extended abstract may be no longer than 5 manuscript sides. Only the first 5 sides of your submission will be sent to the Committee. The schedule for reviewing submissions doesn't permit reviewers to read full papers. You may attach the full paper to the extended abstract. It will not be sent to the Committee but may be helpful during final evaluation.

Every submission should include one additional side stating:

- The name, mail address, daytime and evening phone numbers, e-mail address and (if available) fax number of one of the authors, who will act as the contact point.
- An indication of which, if any, of the authors are students.
- A list of audio/visual equipment desired beyond a microphone and an overhead projector.

Authors of accepted submissions will be notified by May 26, 1993. They will receive instructions for preparing camera-ready copy of the final paper, which must be received by July 8, 1993.

Enquiries about submissions may be made by e-mail to grob@usenix.org or to +33 1 30 64 82 00.

FOR REGISTRATION INFORMATION

Materials containing all details of the symposium program, symposium registration fees and forms, and hotel discount and reservation information will be mailed June 1993. If you wish to receive registration materials, please contact:

- **USENIX Conference Office**
22672 Lambert Street, Suite 613
El Toro, CA 92630 USA
(714) 588-8649; FAX: (714) 588-9706
E-mail: conference@usenix.org

4th

SYMPOSIUM ON EXPERIENCES WITH DISTRIBUTED AND MULTIPROCESSOR SYSTEMS (SEDMS IV)

SEPTEMBER 23-24, 1993
HILTON HOTEL,
SAN DIEGO, CALIFORNIA

IMPORTANT DATES

Dates for Refereed Paper Submissions

- **Submissions due:**
April 27, 1993
- **Notifications mailed:**
June 14, 1993
- **Camera-ready final papers due:**
July 20, 1993

Registration Materials Available:

- July, 1993

ANNOUNCEMENT/CALL FOR PARTICIPATION

Sponsored by: **USENIX**, the USENIX Association
In cooperation with:

- **acm**. ACM SIGARCH, SIGCOMM, SIGOPS and SIGSOFT (Pending)
- **IEEE COMPUTER SOCIETY** IEEE-CS Technical Committees on Distributed Processing, Operating Systems, Software Engineering, and Design Automation (Pending)

The goal of this symposium is to bring together individuals who have built, are building, or will soon build distributed and multiprocessor systems. SEDMS IV provides a forum for individuals to exchange information on their experiences, both good and bad, including experiences with coding aids, languages, debugging and testing technology, reuse of existing software, and performance analysis. The presentations should emphasize the lessons learned from use of such systems and tools.

Papers that have been formally reviewed and accepted will be presented during the symposium and published in the proceedings. Invited talks will complement the peer-reviewed paper presentations. There will also be discussion panels on submitted themes. Extra-long breaks between sessions and works-in-progress reports will be provided to facilitate a workshop-like atmosphere during parts of the symposium.

SUBMISSIONS

Six copies of each submission or panel proposal should be sent to the Program Chair (address below) to arrive no later than April 27, 1993.

Submissions of full papers are invited on any topics related to the theme of the symposium. The committee will give preferential consideration to submissions describing experiences with actual systems. Papers describing purely theoretical work will not be accepted. Panel proposals should include a description of the relevance to the goals of SEDMS and the qualifications of the participants suggested.

For program information, contact:

- | | |
|--|--|
| ➤ General Chair: Peter Reiher
Computer Science Department
Boelter Hall
UCLA
Los Angeles, CA 90024
(310) 206-8696
reiher@wells.cs.ucla.edu | ➤ Program Chair: David Cohn
Computer Science and
Engineering Department
University of Notre Dame
Notre Dame, IN 46556
(219) 239-6694
dlc@cse.nd.edu |
|--|--|

(continued on reverse side)

P R O G R A M
C O M M I T T E E

➤ **General Chair:**

Peter Reiher,
University of California, Los Angeles

➤ **Program Chair:**

David Cohn,
University of Notre Dame

John R. Nicol, *GTE Laboratories, Inc.*

Volker Tschammer, *GMD FOKUS Berlin*

Dag Jo

Karsten Schwan, *Georgia Institute of Technology*

Partha Dasgupta, *Arizona State University*

Brett Fle

David Pitts, *University of Massachusetts, Lowell*

Debra Hensgen, *University of Cincinnati*

John Barr, *Motorola*

Marc Pucci, *Bellcore*

Michael Scott, *University of Rochester*

Mike O'Dell, *Bellcore*

Roy Campbell, *University of Illinois*

Ed Lazowska, *University of Washington*

USENIX

FOR REGISTRATION INFORMATION

Materials containing all details of the symposium program, symposium registration fees and forms, and hotel discount and reservation information will be mailed beginning July 1993. If you wish to receive the registration materials, please contact:

➤ **USENIX Conference Office**

22672 Lambert Street, Suite 613

El Toro, CA 92630 USA

(714) 588-8649

FAX: (714) 588-9706

E-mail: conference@usenix.org

USENIX, the UNIX and Advanced Computing Systems
Professional and Technical Association.

4th USENIX UNIX SECURITY SYMPOSIUM

OCTOBER 4-7, 1993
SANTA CLARA
MARRIOTT HOTEL
SANTA CLARA,
CALIFORNIA

IMPORTANT DATES

Dates for Refereed Paper Submissions:

- Extended abstracts due: June 4, 1993
- Program Committee decisions made: June 30, 1993
- Camera-ready final papers due: August 15, 1993

Registration Materials Available:

- July, 1993



SPONSORED BY THE USENIX ASSOCIATION

In cooperation with:

- The Computer Emergency Response Team (CERT)
- The Association for Computer Machinery (pending)

The goal of this symposium is to bring together security practitioners, system administrators, system programmers, and others with an interest in computer security as it relates to networks and the UNIX operating system.

This will be a three and one-half day, single-track symposium. The symposium will consist of tutorials, refereed and invited technical presentations, and panel sessions. The first day will be devoted to tutorial presentations, followed by two and one-half days of technical sessions. There will also be two evenings available for Birds-of-a-Feather sessions and Works-in-Progress sessions.

TUTORIALS

October 4, 1993

This one-day tutorial program will feature two tutorials, designed to address the needs of both management and technical attendees. The tutorials will supply overviews of various security mechanisms and policies. Each will provide specifics to the system and site administrator for implementing numerous local and network security precautions, firewalls, and monitoring systems.

TECHNICAL SESSIONS

In addition to refereed paper presentations, the program will include invited talks and panel sessions. The program committee invites you to submit proposals, ideas, or suggestions for these presentations

Papers that have been formally reviewed and accepted will be presented during the symposium and published in the symposium proceedings. Proceedings will be distributed free to technical sessions attendees during the symposium and after will be available for purchase from the USENIX Association.

SYMPOSIUM TOPICS

Papers are being solicited in areas including but not limited to:

- User/system authentication
- File system security
- Network security
- Security and system management
- Security-enhanced versions of the UNIX operating system
- Security tools
- Network intrusions (including case studies and intrusion detection efforts)
- Security on high-bandwidth networks

(continued on reverse side)

PROGRAM COMMITTEE

➤ Program Chair:

Bill Cheswick,
AT&T Bell Laboratories
Steve Bellovin,
AT&T Bell Laboratories
Matt Bishop,
Dartmouth College
Ed DeHart,
*CERT, Carnegie Mellon
University*
Jim Ellis,
*CERT, Carnegie Mellon
University*
Marcus Ranum,
Trusted Information Systems

TO SEND SUBMISSIONS

- Send ASCII or Postscript submissions to: ches@research.att.com
- Send hard copy submissions to the program chair:
Bill Cheswick
AT&T Bell Laboratories
Room 2c416
600 Mountain Ave.
Murray Hill, NJ 07974

DATES FOR REFEREED PAPER SUBMISSIONS

- Extended abstracts due: June 4, 1993
- Program Committee decisions made: June 30, 1993
- Camera-ready final papers due: August 15, 1993

FOR REGISTRATION INFORMATION

Materials containing all details of the symposium program, symposium registration fees and forms, and hotel discount and reservation information will be mailed beginning July 1993. If you wish to receive the pre-registration materials, please contact:

- **USENIX Conference Office**
22672 Lambert Street, Suite 613
El Toro, CA 92630 USA
(714) 588-8649; FAX: (714) 588-9706
E-mail: conference@usenix.org

USENIX

USENIX, the UNIX and Advanced Computing Systems Professional and Technical Association.

1993 USENIX SYMPOSIUM ON MOBILE & LOCATION- INDEPENDENT COMPUTING

AUGUST 2-3, 1993
MARRIOTT HOTEL
CAMBRIDGE,
MASSACHUSETTS

IMPORTANT DATES

Dates For Refereed Paper Submissions:

- April 19, 1993
Extended abstracts due
- May 3, 1993
Notification to authors
- June 14, 1993
Camera-ready final papers due

Registration Materials Available:

- May, 1993



Much of the growth of UNIX has been due to its support for casual communications, thus fostering cooperative work within a location-independent framework. The latest incarnation of location independence is "Mobile Computing."

Distributed computing, now fashionable in other circles, was pioneered by the UNIX community. Support for Mobile Computing is the next logical step in assuring the role of UNIX as the operating system that offers a rich and complete feature set.

Progress in Mobile Computing is everywhere evident both in academic and non-academic circles. We intend to concentrate on it in a true state-of-the-art symposium and technical free-for-all on what it takes to make Mobile Computing work and work right.

SYMPOSIUM SCHEDULE

This is a single track symposium offering two days of refereed paper presentations (scheduled for 30 minutes each). The symposium will also include two panels, Works-in-Progress reports, and a Keynote. Birds-Of-a-Feather sessions will take place in the evenings.

Formally reviewed papers, presented during the symposium, will be published in the symposium proceedings. Proceedings will be distributed free to attendees during the symposium and later will be available for purchase from the USENIX Association.

SYMPOSIUM TOPICS

As is usual for a USENIX symposium, we are looking for new and arresting developments in systems that directly contribute to a technical understanding of Mobile Computing. UNIX will be the lingua franca of discussion, but we are eager for progress from other world views to be presented as well. The Mobile Computing Symposium will address a wide range of issues and ongoing developments, including, but not limited to:

- Naming (e.g. Prospero or OSF/DCE DNS)
- Wide area information distribution (e.g. WAIS and archie)
- Security (e.g. authentication based on devices and digital signature services)
- User locatability (e.g. paging systems and active badges)
- Rendezvous (e.g. videoconferencing over the internet and various groupware efforts)
- Networking and Connectability (e.g. the new IETF routing work, movement of "sockets" from site to site, and the rumored advent of IP connections from airplanes)
- Portable tiny devices (e.g. the various palmtops and personal information assistants)

(continued on reverse side)

P R O G R A M
C O M M I T T E E

➤ **General Chair:**

Peter Reiher,
University of California, Los Angeles

➤ **Program Chair:**

David Cohn,
University of Notre Dame

John R. Nicol, *GTE Laboratories, Inc.*

Volker Tschammer, *GMD FOKUS Berlin*

Dag Jo

Karsten Schwan, *Georgia Institute of Technology*

Partha Dasgupta, *Arizona State University*

Brett Fle

David Pitts, *University of Massachusetts, Lowell*

Debra Hensgen, *University of Cincinnati*

John Barr, *Motorola*

Marc Pucci, *Bellcore*

Michael Scott, *University of Rochester*

Mike O'Dell, *Bellcore*

Roy Campbell, *University of Illinois*

Ed Lazowska, *University of Washington*

FOR REGISTRATION INFORMATION

Materials containing all details of the symposium program, symposium registration fees and forms, and hotel discount and reservation information will be mailed beginning July 1993. If you wish to receive the registration materials, please contact:

➤ **USENIX Conference Office**

22672 Lambert Street, Suite 613

El Toro, CA 92630 USA

(714) 588-8649

FAX: (714) 588-9706

E-mail: conference@usenix.org

USENIX

USENIX, the UNIX and Advanced Computing Systems
Professional and Technical Association.

7th USENIX SYSTEMS ADMINISTRATION CONFERENCE (LISA VII)

NOVEMBER 1-5, 1993
MARRIOTT HOTEL
MONTEREY, CALIFORNIA

IMPORTANT DATES

Dates for Refereed Paper Submissions:

- Extended Abstract Submission
Deadline: July 2, 1993
- Notification to Authors:
July 23, 1993
- Final Papers Receipt Deadline:
September 7, 1993

Registration Materials Available:

- August, 1993



ANNOUNCEMENT/CALL FOR PARTICIPATION

The annual LISA conference provides a forum in which system administrators meet to share ideas and experiences. A growing success for the past six years, LISA is the only conference which focuses specifically on the needs of system administrators. Its scope includes system administrators from sites of all sizes and configurations.

TUTORIAL PROGRAM

Monday and Tuesday, November 1-2, 1993

The tutorial program at LISA VII is divided into three tracks with a total of twelve half-day tutorials. Attendees may move between tracks, choosing which sections most interest them. Tutorials offer expert instruction in areas of interest to system administrators, novice through experienced. Topics are expected to include Networking, Programming in Perl, X and the Administrator, the Domain Name System, Sendmail, and more.

TECHNICAL SESSIONS

Wednesday through Friday, November 3-5, 1993

"The Human Aspect of UNIX System Administration" is the theme of the first track of the conference technical sessions. Although papers of a more traditional technical content are also very welcome, the committee is especially seeking papers on areas such as creating policies and procedures and interacting with management and/or users, or which describe and evaluate methods aimed at improved communication with users, and/or management. We are interested in papers which provide freely available or fully described solutions to existing problems.

The second track of the conference technical sessions will be split between presentations on very large installation system administration and presentations of practical, experience-derived material of special interest to new system administrators.

No simple measure defines "large installation." It could be number of hosts, number of users, size of network, amount of on-line storage, or some combination of these. The only certainty is that today's large will be tomorrow's standard. We wish to hear from sites which have unique problems and solutions relating to the management of large installations. We seek proposals for papers, panels, mini-workshops, or similar presentations for this track.

We also seek papers, mini-workshops, or panel presentations of pragmatic material from experienced system administrators who wish to share their experiences, hardships and solutions. It is hoped that administrators at all levels can leverage this track to solve specific problems at their site.

VENDOR DISPLAY

Wednesday, November 3, 1993, 5:00 pm-9:00 pm

Well informed vendor representatives will demonstrate products and services useful to system and network administration at the informal table-top display accompanying the LISA Conference. If your company would like to participate, please contact Cynthia Deno at (408) 335-9445, FAX (408) 335-2163, E-mail: cynthia@usenix.org

CONFERENCE TOPICS

The technical sessions program may include invited talks, panels, Works-In-Progress (WIP) reports, and Birds-Of-a-Feather (BOF) sessions, alongside the refereed paper presentations. The program committee invites you to submit informal proposals, ideas, or suggestions on any of the following or related topics:

- Implementation, usage, and strategies for Policies and Procedures
- Effects of improved communication with users and/or management.
- Tricks in user education
- How to develop Junior System Administrators
- System Security Monitoring
- Security issues, especially where multiple people are privileged users
- Heterogeneous system administration
- Human issues of administration
- Graphical User Interfaces for system administration

(continued on reverse side)

DATES FOR REFEREED PAPER SUBMISSIONS

- Extended Abstract Submission
Deadline: July 2, 1993
- Notification to Authors:
July 23, 1993
- Final Papers Receipt Deadline:
September 7, 1993

ADDRESS FOR SUBMISSIONS

For submission of all proposals
and extended abstracts of refereed
papers, and for program informa-
tion, contact:

- **Program Chair:**
Bjorn Satdeva
2787 Moorpark Avenue
San Jose, CA 95128
(408) 241-3111
E-mail: bjorn@sysadmin.com

FOR REGISTRATION INFORMATION

Materials containing all details of the
symposium program, symposium
registration fees and forms, and
hotel discount and reservation infor-
mation will be mailed and posted to
the net beginning August 1993. If
you wish to receive registration
materials, please contact:

- **USENIX Conference Office**
22672 Lambert Street, Suite 613
El Toro, CA 92630 USA
(714) 588-8649
FAX: (714) 588-9706
E-mail: conference@usenix.org

USENIX

- Distributed system administration
- Network growth and performance management
- Network management
- Network monitoring
- Wireless LANs
- Evaluating performance of high-end workstations and servers
- Integration of heterogeneous systems
- Usage monitoring and accounting systems
- Administration of remote sites

REFEREED PAPER SUBMISSIONS

The committee requires that an extended abstract be submitted for the paper selection process. (Full-papers are not acceptable for this stage; if you send a full paper, you must also include an extended abstract for evaluation.) Your extended abstract should consist of a traditional abstract which summarizes the content/ideas of the entire paper, followed by a skeletal outline of the full paper. We require electronic (nroff/troff, TeX or ASCII) submission of the extended abstract.

Authors of an accepted paper will present their paper at LISA VII and provide a final paper for publication in the *Conference Proceedings*. Final papers are limited to 20 pages, including diagrams, figures and appendix. Papers should include a brief description of the site (if applicable), an outline of the problem and issues, and details of the solution. Authors may provide electronic versions or camera-ready copy (instructions will be provided upon request) of final papers. Conference proceedings will be distributed to all conference attendees and will also be available from the USENIX Association after the conference.

CONFERENCE COMMITTEE

- **Program Chair:** Bjorn Satdeva, */sys/admin, Inc.*
Brent Chapman, *Great Circle Associates*
Lee Damon, *Castle PAUS*
Tina M. Darmohray, *Lawrence Livermore National Labs*
Janet Frazer, *UNIX System Laboratories, Inc.*
Helen Harrison, *SAS Institute*
Dinah McNutt, *Tivoli Systems*
Bryan McDonald, *SRI International*
Arch Mott, *Cisco Systems, Inc.*
Paul Moriarty, *Cisco Systems, Inc.*
Jeff Polk, *Berkeley Software Design, Inc.*
Greg Rose, *Australian Computing and Communications Institute*
Peg Schafer, *Bolt Beranek & Newman, Inc.*
Steve Simmons, *Industrial Technology Institute*
Liza Y. Weissler, *RAND Corporation*
Pat Wilson, *Dartmouth College*
Elizabeth Zwicky, *SRI International*

USENIX, the UNIX and Advanced Computing Systems
Professional and Technical Association.

1993

WORLD CONFERENCE ON TOOLS AND TECHNIQUES FOR SYSTEM ADMINISTRATION, NETWORKING, AND SECURITY (SANS-II)

APRIL 19-23, 1993

STOUFFERS

CONCOURSE HOTEL,
ARLINGTON, VIRGINIA

CO-SPONSORED BY THE
USENIX ASSOCIATION,
SAGE (THE SYSTEM
ADMINISTRATORS'
GUILD), AND FEDUNIX



REGISTRATION ANNOUNCEMENT

SANS-II, the second international conference combining system administration, network management, and security in UNIX computing environments, offers authoritative tutorial courses and a forum in which system administrators, network managers, and security experts can exchange practical information, share new ideas, and evaluate new tools. SANS-II expands on the highly-acclaimed 1992 World Conference on System Administration and Security by including network management issues. SANS-II continues the tradition with its exclusive focus on practical, cost-effective solutions for today's problems. It also provides the opportunity to review the newest support tools and to focus on how these tools can interact to lower the costs of managing distributed computing.

CONFERENCE SESSIONS OVERVIEW

Tutorial Courses: April 19-20, 1993

The conference begins with two days of courses featuring more than a dozen full-day courses taught by several of America's top-rated instructors. Tutorial offerings include: UNIX System Internals, UNIX Fundamentals, OSF DCE (two days), OSF DME, Basic UNIX Security, Advanced UNIX Security, Practical PERL Programming, Introduction to TCP/IP, UNIX Network Programming, UNIX System Administration, UNIX Network Administration, Advanced Topics in System Administration Part I, and Advanced Topics in System Administration: Part II.

Technical Sessions: April 21-23, 1993

During the three days of technical sessions, peer-reviewed papers will be complemented by invited papers. Papers that have been formally reviewed and accepted will also be published in the conference proceedings. The Papers Review Committee is composed of experts on system administration, network management, and security along with managers of large installations and architects from the vendor community.

In the special "Tools Track" delegates will learn, through technical briefings, how the most important commercial systems and network management software and hardware products actually work. This track, which runs simultaneously with the papers track, is especially helpful to organizations that are attempting to move toward commercial off-the-shelf (COTS) software for systems management in UNIX computing environments.

Additional technical sessions brought back by popular demand:

- ◆ "Ask the Experts" sessions where you'll find practical answers to your questions.
- ◆ "Best of the Net" session where you'll learn which free programs available from the net are most useful.
- ◆ "Ask OSF" where you can learn from the people who brought you DCE and are bringing you DME.
- ◆ Informal **Birds-of-a-Feather** sessions in the evening to allow for additional topics. Please send your suggestions for topics with your registration.
- ◆ "Tips and Techniques" sessions in which conference attendees can share, in 5-minute presentations, their favorite techniques for solving recurring problems. These sessions are run as moderated BOFs with all conference attendees being asked, in advance, to contribute if they choose.

WHY YOU SHOULD PARTICIPATE

Growing, heterogeneous networks add complexity to every system administration task and increase the risk of security breaches. Increased expectations make systems and network management far more difficult and time-consuming. You are pressured to do more with fewer people. These challenges are particularly frustrating in government agencies, universities, and companies which have been in the vanguard of the move to open systems and networks of UNIX computers.

At SANS-II you will explore with your colleagues countermeasures to these challenges which are making your job more and more difficult. And at SANS-II, you can see the best of the new commercial automated management tools. The recent emergence of these tools promises to reduce the effort and lower the cost of complex system administration. Here's your opportunity to find where and how they can help you.

(continued on reverse side)

CONFERENCE LOCATION

- **Stouffers Concourse Hotel**
(at National Airport)
2399 Jefferson Davis Highway
Arlington, Virginia 22202
Telephone: (703) 418-6800

PAPERS REVIEW COMMITTEE

- **Papers Chairman:**
Alan Paller, *FedUNIX*
Matt Bishop, *Dartmouth College*
Bryan McDonald, *SRI International*
Paul Moriarty, *Cisco*
Marcus Ranum, *Digital Equipment Corporation*
Tom Christiansen, *CONVEX Computer Corporation*
Dave Brillhart, *Harris Semiconductor*
Elizabeth Zwicky, *SRI International*
Norman Kincl, *Hewlett Packard*
Michele Crabb, *NASA Ames*
Bruce Hunter, *Intel*
Dale Pfaff, *US Naval Research Laboratory*
William Howell, *University of North Carolina*
Rob Kolstad, *Berkeley Software Design, Inc.*

FOR MORE INFORMATION

- **Registration Information and Materials**
Please contact:
Conference Office
FedUNIX
4610 Tournay Road
Bethesda, MD 20816
Telephone: (301) 229-1062
Email: sans@fedunix.org

SANS-II is designed to identify the state of the art for cost-effective systems and network administration and security. In this way the techniques and tools used by the most effective managers can be adopted by those still looking for solutions. System administrators, security administrators, network administrators, network managers, technology managers, computer installation managers, and their staff should attend.

CONFERENCE TOPICS:

- Making Backup Less Painful
- Mail Handling
- Automating Console Operations
- System Scheduling and Monitoring
- Better Storage Solutions
- Accounting and Chargeback
- Off-The-Shelf Tools
- Solutions that Caused Problems
- Security and Audit Management
- Security Policies
- Policies and Procedures On the Network
- OS's DCE and DME
- Training and Education
- Techniques For Dealing With Users
- Managing Heterogeneous Networks
- Network Security Monitoring
- Network Monitoring and Performance Testing
- Networked Backup Schemes
- Distributed Mail Systems
- Domain Name Service Configuration
- Distributed Console Access

REGISTRATION FEES:

		Technical Sessions Only	Technical Sessions + 1 Tutorial	Technical Sessions + 2 Tutorials
.....				
Alumni of The				
1992 World	<i>Before 2/15/93</i>	\$325	\$595	\$775
Conference	<i>After 2/15/93</i>	\$395	\$670	\$895
On System	<i>After 3/24/93</i>	\$445	\$730	\$945
Administration				
and Security or				
of The 1992 LISA				
Conference				
.....				
	<i>Before 3/24/93</i>	\$395	\$670	\$895
All Others	<i>After 3/24/93</i>	\$445	\$730	\$945

UPCOMING SYMPOSIA AND CONFERENCES



APRIL 19-21, 1993

3RD MACH SYMPOSIUM

Program Chair: David Black,
Open Software Foundation

El Dorado Hotel,
Santa Fe, New Mexico



APRIL 19-23, 1993

1993 WORLD CONFERENCE ON TOOLS AND TECHNIQUES FOR SYSTEM ADMINISTRATION, NETWORKING AND SECURITY (SANS-II)

Papers Chairman: Alan Paller, *FedUNIX*

Stouffers Concourse Hotel,
Arlington, Virginia

Co-sponsored with SAGE, the Systems Administrators' Guild,
and FedUNIX

JUNE 21-25, 1993

SUMMER 1993 GENERAL CONFERENCE

Program Chair: David S. H. Rosenthal,
SunSoft, Inc.

Cincinnati Convention Center,
Cincinnati, Ohio



AUGUST 2-3, 1993

SYMPOSIUM ON MOBILE & LOCATION-INDEPENDENT COMPUTING

Program Chair: Dan Geer,
Geer Zolot Associates

Vice-Program Chair: Clement Cole,
Locus Computing Corporation

Marriott Hotel,
Cambridge, Massachusetts



SEPTEMBER 20-22, 1993

2ND SYMPOSIUM ON MICROKERNELS & OTHER KERNEL ARCHITECTURES

Program Chair: Lori S. Grob,
Chorus systèmes

Hilton Beach & Tennis Resort,
San Diego, California



SEPTEMBER 23-24, 1993

EXPERIENCES WITH DISTRIBUTED & MULTIPROCESSOR SYSTEMS (SEDMS IV)

General Chair: Peter Reiher, *UCLA*

Program Chair: David Cohn,
University of Notre Dame

Hilton Beach & Tennis Resort,
San Diego, California

In cooperation with: ACM SIGARCH, SIGCOMM, SIGOPS and SIGSOFT
and IEEE-CS Technical Committees on Distributed Processing,
Operating Systems, Software Engineering, and Design Automation

OCTOBER 4-7, 1993

4TH UNIX SECURITY SYMPOSIUM

Co-sponsored with The Computer
Emergency Response Team (CERT)

Program Chair: Bill Cheswick,
AT&T Bell Laboratories

Santa Clara Marriott Hotel,
Santa Clara, California



NOVEMBER 1-5, 1993

7TH SYSTEMS ADMINISTRATION CONFERENCE (LISA VII)

Co-sponsored with SAGE,
the Systems Administrators' Guild

Program Chair: Bjorn Satdeva,
/sysladmin, Inc.

Marriott Hotel,
Monterey, California

JANUARY 17-21, 1994

WINTER 1994 TECHNICAL CONFERENCE

Program Chair: Jeffrey Mogul,
Digital Equipment Corporation

Hilton Hotel,
San Francisco, California



POSTPONED

UNIX APPLICATIONS DEVELOPMENT SYMPOSIUM

Co-sponsored by UniForum Canada



ALSO IN 1994

SPRING 1994: 6TH C++ CONFERENCE

Program Chair: Doug Lea,
SUNY Oswego

JUNE 6-10, 1994: USENIX SUMMER 1994 TECHNICAL CONFERENCE

Hynes Convention Center
Boston, Massachusetts



TO RECEIVE FULL INFORMATION

Please contact: **USENIX Conference Office, 22672 Lambert St., Suite 613, El Toro, CA 92630 USA**
(714) 588-8649; FAX (714) 588-9706; email: conference@usenix.org

Special Offer: Computing Systems

USENIX and the University of California Press are offering a substantial discount for a limited time on the 1988-1991 volumes (1-4) of *Computing Systems*, as follows:

- 50% off the complete 4 volume set (16 issues) = \$112
- 40% off single whole volumes (4 issues) = \$33.60 per volume
- 35% off single issues = \$9.10

To order back issues please reference the listing of articles on the adjacent page and fill out the form below:

I would like to order: Amount Due

☐ Volumes 1-4, 1988-1991, @ \$112 per volume \$_____

☐ Single volume, number _____ @\$33.60 each
_____ @\$33.60
_____ @\$33.60
_____ @\$33.60

☐ Single issue, vol. _____ num. _____ @\$9.10 each
_____ @\$9.10
_____ @\$9.10
_____ @\$9.10

Order Sub Total _____

Add CA Sales Tax _____

Additional International Postage** ☐ Surface ☐ Air _____

Total amount enclosed \$_____

Payment Options:

_____ Check enclosed payable to *Computing Systems*.

_____ Credit card: _____ Visa _____ MasterCard

Account # _____ Expiration Date _____

Signature _____

**International order? Please add one of the following amounts for postage:

	Surface	Air Freight
Complete 4 Volume Set:	\$23.50	\$65.00
Single Volume:	\$ 5.00	\$36.00
Single Issue:	\$ 1.25	\$ 9.00

Please make your payment in U.S. currency by: a check drawn on a U.S. bank, charge (Visa or MasterCard), or international postal money order.

Ship To: Name _____

Address _____

City _____ State/Country _____ Zip _____

Please send this form along with your payment to:

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710

Fax: 510/548-5738
Phone: 510/528-8649
email: office@usenix.org

VOLUME 1

Number 1, Winter 1988: "The Synthesis Kernel", Pu, Massalin, Ioannidis
"Language and Operating System Features for Real-time Programming", Donner, Jameson
"Dynamics for Computer Graphics: A Tutorial", Wilhelms

Number 2, Spring 1988: "Enhanced Resource Sharing in UNIX", Barton, Wagner
"Design and Implementation of Parallel Make", Baalbergen
"Yacc Meets C++", Johnson
"Watchdogs - Extending the UNIX File System", Bershad, Pinkerton
"Controversy: Can UNIX survive secret source code?", Lesk

Number 3, Summer, 1988: "GRAB - Inverted Indexes with Low Storage Overhead", Lesk
"An Application of a Fast Data Encryption Standard Implementation", Bishop
"Effects of a copy-on-write Memory Management on the Response Time of UNIX fork Operations", Smith, Maguire
"Controversy: Window Systems Should be Transparent", Pike

Number 4, Fall 1988: "CHORUS Distributed Operating Systems", Rozier, et al.
"Type-safe Linkage for C++", Stroustrup
"An Unorthodox Approach to Undergraduate Software Engineering Instruction", Morris

VOLUME 2

Number 1, Winter 1989: "Developing Applications for Heterogeneous Machine Networks: The Durra Environment", Barbacci, et al.
"A Hypertext System for UNIX", Brown
"Parameterized Types for C++", Stroustrup

Number 2, Spring 1989: "Page Makeup by Postprocessing Text Formatter Output", Kernighan, Van Wyk
"A Concurrent Window System", Pike
"Experience with Viruses on UNIX Systems", Duff
"Virology 101", McIlroy

Number 3, Summer 1989: "The Evolution of C++: 1985 to 1989", Stroustrup
"Heuristics for Disk Drive Positioning in 4.3BSD", Stevens

Number 4, Fall 1989: "SOS: An Object-Oriented Operating System - Assessment and Perspectives", Shapiro, et al.
"Data Structures in the Icon Programming Language", Griswold
"Multiple Inheritance for C++", Stroustrup

VOLUME 3

Number 1, Winter 1990: "The Design and Implementation of the Clouds Distributed Operating System", Dasgupta, et al.
"Using Hints in DUNE Remote Procedure Calls", Pucci, Alberi
"Mach/4.3BSD: A Conservative Approach to Parallelization", Boykin, Langerman
"Implementation Issues for the Psyche Multiprocessor Operating System", Scott, et al.
"Fine-Grain Adaptive Scheduling using Feedback", Massalin, Pu

Number 2, Spring 1990: "Little Languages for Music", Langston
"The Personal Orchestra, or Audio Data Compression by 10,000:1", Hawley
"Keynote - A Language and Extensible Graphic Editor for Music", Thompson
"Controversy: Portability - A No Longer Solved Problem", Feldman, Gentleman

Number 3, Summer 1990: "Process Synchronization in the UTS Kernel", Ruane
"A Concurrent Programming Support for Distributed Systems", Spezzano, Talia, Vanneschi
"Distributed Spooling in a Heterogeneous Environment", Wagner

Number 4, Fall 1990: "An Experimental Implementation of the Tilde Naming System", Comer, Droms, Murtagh
"An Object Model for Conventional Operating Systems", Dewan, Vasilik
"A Comparison of Basic CPU Scheduling Algorithms for Multiprocessor UNIX", Curran, Stumm

VOLUME 4

Number 1, Winter 1991: "A System for Algorithm Animation", Bentley, Kernighan
"Architecture and Implementation of Guide, and Object-Oriented Distributed System", Balter, et al.
"Controversy: The Case Against Multiple Inheritance in C++", Cargill

Number 2, Spring 1991: "expect: Scripts for Controlling Interactive Processes", Libes
"An ASCII Database for Fast Queries of Relatively Stable Data", Herrin, Finkel
"Controversy: The Case for Multiple Inheritance in C++", Waldo

Number 3, Summer 1991: "Experience Developing the RP3 Operating System", Byrant, Chang, Rosenberg
"Configurable Data Manipulation in an Attached Multiprocessor", Pucci
"Distributed Programming with Objects and Threads in the Clouds System", Dasgupta, et al.
"Evolution of a Communication System for Distributed Transaction Processing in Raid", Bhargava, Zhang, Mafla
"Measured Performance of Caching in the Sprite Network File System", Welch

Number 4, Fall 1991: "A Comparison of Two Distributed Systems: Amoeba and Sprite", Douglass, et al.
"The Software Design Laboratory", Smith
"Swift: Using Distributed Disk Striping to Provide High I/O Data Rates", Cabrera, Long

Publications Order Form



Conference & Workshop Proceedings

Qty	Proceedings		Member Price	Non-Member Price	Subtotal	Overseas Postage	Total
WINTER/SUMMER CONFERENCES							
___	San Diego	Winter '93	33	40	\$ _____	\$25	\$ _____
___	San Antonio	Summer '92	23	30	\$ _____	14	\$ _____
___	San Francisco	Winter '92	30	39	\$ _____	22	\$ _____
___	Nashville	Summer '91	32	38	\$ _____	22	\$ _____
___	Dallas	Winter '91	28	34	\$ _____	18	\$ _____
___	Anaheim	Summer '90	22	22	\$ _____	15	\$ _____
___	Washington, DC	Winter '90	25	25	\$ _____	15	\$ _____
___	Baltimore	Summer '89	20	20	\$ _____	15	\$ _____
___	San Diego	Winter '89	30	30	\$ _____	20	\$ _____
___	San Francisco	Summer '88	29	29	\$ _____	20	\$ _____
___	Dallas	Winter '88	26	26	\$ _____	15	\$ _____
___	Phoenix	Summer '87	35	35	\$ _____	20	\$ _____
___	Washington, DC	Winter '87	10	10	\$ _____	15	\$ _____
___	Atlanta	Summer '86	37	37	\$ _____	20	\$ _____
___	Denver	Winter '86	25	25	\$ _____	15	\$ _____
___	Portland	Summer '85	45	45	\$ _____	25	\$ _____
___	Dallas	Winter '85	15	15	\$ _____	10	\$ _____
___	Salt Lake City	Summer '84	29	29	\$ _____	20	\$ _____
___	Washington, DC	Winter '84	25	25	\$ _____	15	\$ _____
___	Toronto	Summer '83	32	32	\$ _____	20	\$ _____
___	San Diego	Winter '83	28	28	\$ _____	15	\$ _____
LARGE INSTALLATION SYSTEMS ADMINISTRATION							
___	LISA VI	Oct. '92	23	30	\$ _____	12	\$ _____
___	LISA V	Sept. '91	20	23	\$ _____	11	\$ _____
___	LISA IV	Oct. '90	15	18	\$ _____	8	\$ _____
___	LISA III	Sept. '89	13	13	\$ _____	9	\$ _____
___	LISA II	Nov. '88	8	8	\$ _____	5	\$ _____
___	LISA I	April '87	4	4	\$ _____	5	\$ _____
C++							
___	C++ Conference	Aug. '92	30	39	\$ _____	20	\$ _____
___	C++ Conference	Apr. '91	22	26	\$ _____	11	\$ _____
___	C++ Conference	Apr. '90	28	28	\$ _____	18	\$ _____
___	C++ Conference	Oct. '88	30	30	\$ _____	20	\$ _____
___	C++ Workshop	Nov. '87	30	30	\$ _____	20	\$ _____
SECURITY							
___	UNIX Security III	Sept. '92	30	39	\$ _____	11	\$ _____
___	UNIX Security II	Aug. '90	13	16	\$ _____	8	\$ _____
___	UNIX Security	Aug. '88	7	7	\$ _____	5	\$ _____
MACH							
___	Mach Symposium	Nov. '91	24	28	\$ _____	14	\$ _____
___	Mach Workshop	Oct. '90	17	20	\$ _____	9	\$ _____



Qty	Proceedings		Member Price	Non-Member* Price	Subtotal	Overseas Postage	Total
DISTRIBUTED & MULTIPROCESSOR SYSTEMS (SEDMS)							
___	SEDMS III	Mar. '92	30	36	\$ _____	20	\$ _____
___	SEDMS II	Mar. '91	30	36	\$ _____	20	\$ _____
___	SEDMS	Oct. '89	30	30	\$ _____	20	\$ _____
GRAPHICS							
___	Graphics Workshop V	Nov. '89	18	18	\$ _____	10	\$ _____
___	Graphics IV	Oct. '87	10	10	\$ _____	10	\$ _____
___	Graphics III	Nov. '86	10	10	\$ _____	5	\$ _____
___	Graphics II	Dec. '85	7	7	\$ _____	5	\$ _____
OTHER WORKSHOPS							
___	File Systems	May '92	15	20	\$ _____	9	\$ _____
___	Micro-Kernel & Other Kernel Arch.	April '92	30	39	\$ _____	20	\$ _____
___	UNIX Transaction Processing	May '89	12	12	\$ _____	8	\$ _____
___	Software Management	Apr. '89	20	20	\$ _____	15	\$ _____
___	UNIX & Supercomputers	Sept. '88	20	20	\$ _____	10	\$ _____

Discounts are available for bulk orders. Please inquire.

Total price of Proceedings _____ **
Calif. residents add sales tax _____
Total overseas postage _____
Total enclosed _____

**If you are paying member price, please include member's name and/or
 membership number _____

PAYMENT OPTIONS*

___ Check enclosed - payable to USENIX Association
 ___ Charge my: ___ VISA  ___ MC  Account # _____ Exp. Date _____
 ___ Purchase order enclosed Signature _____

* Outside the USA? Please make your payment in US currency by one of the following:

- Check - issued by a local branch of a US Bank
- Charge (VISA, MasterCard, or foreign equivalent)
- International postal money order

Shipping Information

Ship to: _____

Please allow 2-3 weeks for delivery. Overseas orders are shipped via air printed matter.

Please mail or fax this order form with payment to:

USENIX Association
 2560 Ninth Street, Suite 215
 Berkeley, CA 94710
 FAX 510/548-5738

- If you are not a member and wish to receive our membership information packet, please check this box. ☐

Membership Application



MEMBERSHIP INFORMATION

Any individual or institution may become a member of the USENIX Association by filling out an application form and paying the appropriate annual fee.

There are five classes of membership:

Student: \$20
Open to any full-time student at an accredited educational institution. A copy of the current student I.D. card must be provided.

Individual: \$65
Open to any individual or institution. Individual Members may vote.

Corporate: \$325
Corporate Membership is open to any individual or institution.

Educational: \$160
Educational Membership is open to accredited educational institutions.

Supporting: \$1000
Open to any individual or institution that wants to support the Association to a greater degree than through the Corporate Membership fee.

Corporate, Educational and Supporting members have one designated representative who receives all services available to Individual Members, plus copies of the proceedings from all conferences and workshops that are held during the term of membership.

SAGE, the Systems Administrators' Guild

USENIX recently launched its first Special Technical Group, the Systems Administrators' Guild (SAGE). SAGE is devoted to the advancement of systems administration as a profession. It will recruit talented individuals to the profession, develop guidelines for the education of members of the profession, establish standards of professional excellence and provide recognition for those who attain them, and promote work that advances the state of the art and propagates knowledge of good practice in the profession.

USENIX and SAGE will work together to publish technical information and sponsor conferences, symposia, tutorials and local groups in the field of systems administration. SAGE news and other items of interest to systems administrators are found in each issue of the USENIX newsletter ;login:. SAGE members must also be USENIX members.

MEMBERSHIP APPLICATION

New ☐ Renewal ☐

Name

Address

City State Zip Country

Phone email address:

- | | | |
|---|--|---|
| <input type="checkbox"/> \$20 Student (full-time)
(with copy of I.D. card) | <input type="checkbox"/> \$160 Educational Institution | <input type="checkbox"/> \$25 - SAGE (Additional) |
| <input type="checkbox"/> \$65 Individual | <input type="checkbox"/> \$325 Corporate | |
| | <input type="checkbox"/> \$1000 Supporting | |

PAYMENT OPTIONS

- ☐ My total amount for membership is \$.
☐ Check enclosed payable to USENIX Association.
☐ Please charge my: ☐ Visa ☐ MasterCard ☐ Purchase order enclosed (Educational and Corporate members only).

Account #   Exp. Date

Signature

Outside the U.S.A.? Please make your payment in U.S. currency by one of the following:

- * Charge (Visa, MasterCard, or foreign equivalent)
- * International postal money order
- * Check - issued by a local branch of a U.S. Bank

1/93

\$24 of your annual membership dues is for a one-year subscription to the newsletter, ;login:, and \$30 of your dues is for a one-year subscription to the journal, Computing Systems.

USENIX Mailing List

- ☐ I do not want my address made available to other members.
☐ I do not want my address made available for commercial mailings.

For Corporate, Educational, and Supporting Memberships Only

Corporate, Educational, and Supporting Members with UNIX source licenses may take advantage of certain services requiring license verification. Please indicate the type of source license(s) held:

- ☐ System V ☐ System III ☐ UCB 4.x BSD ☐ 32V
☐ UCB 2.x BSD ☐ V7 ☐ Other: _____

If you do hold UNIX source licenses, please send the signature page and the pages that show the version of UNIX you are licensed for, the name of the institution owning the license, and the type, serial number, and location of the CPUs.

If you hold Berkeley licenses, send copies of those also. If you have more than one license, please send the above information for each.

Corporate, Educational, and Supporting members renewing their memberships need send only newly acquired licenses or those not previously verified by USENIX.

- ☐ License(s) enclosed ☐ License(s) already on file ☐ Tape Release Form enclosed

Authorized Signature: _____ Date: _____

Please return this form with your purchase order or payment to:

USENIX Association
2560 Ninth Street
Suite 215
Berkeley, CA 94710

Local User Groups

The Association will support local user groups by doing a mailing to assist in the formation of a new group and publishing information on local groups in *login*. At least one member of the group must be a current member of the Association. Send additions and corrections to: login@usenix.org.

CA - Fresno:

The Central California UNIX Users Group consists of a uucp-based electronic mailing list to which members may post questions or information. For connection information:

Educational and governmental institutions:
Brent Auernheimer (209) 278-2573
brent@CSUFresno.edu or *csufres!brent*

Commerical institutions or individuals:
Gordon Crumal (209) 251-2648
csufres!gordon

CA - Orange County:

Meets the 2nd Monday of each month

UNIX Users Association of Southern California
Paul Muldoon (714) 556-1220 ext. 137
New Horizons Computer Learning Center
1231 E. Dyer Rd., Suite 140
Santa Ana, CA 92705

CO - Boulder:

Meets monthly at different sites. For meeting schedule, send email to fruug-info@fruug.org.

Front Range UNIX Users Group
Software Design & Analysis, Inc.
1113 Spruce St., Ste. 500
Boulder, CO 80302
Steve Gaede (303) 444-9100
gaede@fruug.org

D.C. - Washington, D.C.:

Meets 1st Tuesday of each month.

Washington Area UNIX Users Group
9811 Mallard Drive
Laurel, MD 20708
Alan Fedder (301) 953-3626

FL - Coral Springs:

S. Shaw McQuinn (305) 344-8686
8557 W. Sample Road
Coral Springs, FL 33065

FL - Western:

Meets 1st Thursday of each month.

Florida West Coast UNIX Users Group
Richard Martino (813) 536-1776
Tony Becker (813) 799-1836
mcrsys!tony
Ed Gallizzi, Ph.D. (813) 864-8272
e.gallizzi@compmail.com
Jay Ts (813) 979-9169
uunet!pdn!tscs!metran!jan
Dave Lewis (407) 242-4372
dhl@ccd.harris.com

FL - Orlando:

Meets the 3rd Thursday of each month.

Central Florida UNIX Users Group
Mikel Manitiuss (407) 444-8448
mike@aaa.com

FL - Melbourne

Meets the 3rd Monday of every month.

Space Coast UNIX User's Group
Steve Lindsey (407) 242-4766
lindsey@vnet.ibm.com

KS or MO - Kansas:

Meets on 2nd Monday of each month.

Kansas City UNIX Users Group (KUUG)
813B Street
Blue Springs, MO 64015
(816) 235-5212
mlg@cstp.umkc.edu

GA - Atlanta:

Meets on the 1st Monday of each month in White Hall, Emory University.

Atlanta UNIX Users Group
P.O. Box 12241
Atlanta, GA 30355-2241
Mark Landry (404) 365-8108

MI - Detroit/Ann Arbor

Meets on the 2nd Thursday of each month in Ann Arbor.

Southeastern Michigan Sun Local Users Group
and Nameless UNIX Users Group
Steve Simmons office: (313)769-4086
home: (313) 426-8981
scs@lokkur.dexter.mi.us

MN - Minneapolis/St. Paul:

Meets the 1st Wednesday of each month.

UNIX Users of Minnesota
17130 Jordan Court
Lakeville, MN 55044
Robert A. Monio (612) 220-2427
pnessutt@dmsmq.mn.org

MO - St. Louis:

St. Louis UNIX Users Group
P.O. Box 2182
St. Louis, MO 63158
Terry Linhardt (314) 772-4762
uunet!jgaltstl!terry

NE - Omaha:

Meets monthly.

/usr/group/nebraska
P.O. Box 31012
Omaha, NE 68132
Phillip Allendorfer (402) 423-1400

New England - Northern:

Meets monthly at different sites.

Peter Schmitt 603) 646-2085
Kiewit Computation Center
Dartmouth College
Hanover, NH 03755
Peter.Schmitt@dartvax.dartmouth.edu

NJ - Princeton:

Meets monthly.

Princeton UNIX Users Group
Mercer County Community College
1200 Old Trenton Road
Trenton, NJ 08690
Peter J. Holsberg (609) 586-4800
mccc!pjh

NM - Albuquerque:

ASIGUNIX meets every 3rd Wednesday of each month.

Phil Hertz 505/275-0466.

NY - New York City:

Meets every other month in Manhattan.

Unigroup of New York City
G.P.O. Box 1931
New York, NY 10116

OK - Tulsa:

Meets 2nd Wednesday of each month.

Tulsa UNIX Users Group, \$USR
Stan Mason (918) 560-5329
tulsix!smason@drd.com
Mark Lawrence (918) 743-3013
mark@drd.com

TX - Austin:

Meets 3rd Thursday of each month.

Capital Area Central Texas UNIX Society
P.O. Box 9786
Austin, TX 78766-9786
officers@cactus.org
Tom Painter (512) 835-5457
president@cactus.org

TX - Dallas/Fort Worth:

Dallas/Fort Worth UNIX Users Group
660 Preston Forest, Suite 177
Dallas, TX 75230
Kevin Coyle (214) 991-5512
kevincd@shared.com

TX - Houston:

Meets 3rd Tuesday of each month.

Houston UNIX Users Group
(Hounix) answering machine (713) 684-6590
Bob Marcum, President (713) 270-8124
Chuck Bentley, Vice-president
(713) 789-8928 chuckb@hounix.uucp

WA - Seattle:

Meets monthly.

Seattle UNIX Group Membership Info.
Bill Campbell (206) 947-5591
6641 East Mercer
Mercer Island, WA 98040-0820
bill@celestial.com

CANADA - Toronto:

143 Baronwood Court
Brampton, Ont. Canada L6V 3H8
Evan Leibovitch (416) 452-0504
evan@telly.on.ca

LISA Groups

Back Bay LISA

Forum covering all aspects of System and Network Administration, for large and small installations. Meets Monthly, various locations in Boston.

JR. Oldroyd
The Instruction Set
601 Trapelo Road
Waltham MA 01254
(617) 890 4930
jr@inset.com

Mailing list: bblisa@inset.com
List Requests: bblisa-request@inset.com

BAY LISA

The Bay-LISA group meets monthly in Santa Clara, CA, to discuss topics of interest for administration of sites with more than 100 users and/or computers.

Send e-mail to baylisa-info@sysadmin.com,
or you may contact:

Bjorn Satdeva
(408) 241-3111
bjorn@sysadmin.com

PrimeTime Freeware

Issue 1-2 of Prime Time Freeware (cover date July, 1992)

The next issue of PTF is in production. Here is a summary of the issue; contact us [<ptf@cfcl.com>](mailto:ptf@cfcl.com) if you want more detailed information:

Format: two ISO-9660 CD-ROMs, bound into a 50+ page booklet. Each disc contains around 1/2 GB of compressed archives, annotation files, etc. The issue unpacks to around 3 GB (3000 megabytes).

Content: PTF is primarily a collection of UNIX-related freeware source code. Binary files and support for non-UNIX platforms are strictly incidental. There isn't room to list everything, but here are some of the bigger items:

ada.xlib, Andrew, ANU NEWS, Athena, btool, CLM, CLU, CLUE, CLX, CMU Common Lisp, comp.sources.{3b1,amiga,games,misc,reviewed,sun,unix,x}, Condor, COOL, CRISP, dirt, Ezd, Epoch, Franz Lisp, GINA, GNU (prep: /pub/gnu/, DJG++, GNUish MSDOS, the Cygnus Solaris-2 and Vintage releases), Go (2D graphics library), Grass, HyperNeWS, Icon (several OS's, plus examples), IMAP, INGRES, InterViews, ISODE, Kermit (tapes A-E), LispView, Lucid Emacs, Mach, MAEstro, magic, MH, NCSA Data Analysis Tools, NIHCL, Oaklisp, PARI,*

PCL, PCLU, Pine, PlaNet, Postie Pat, Q, SCHEME (asst. versions), Scorpion, Serpent, SR, SRC Modula-3, T, Tcl (Tk, expect, etc.), TIFF, TXL, UnixTeX, URT, UIT, VOGL, VOGLE, VOPL, VORT, wframe, WINTERP, WRL Modula-2, X11R5p13, XView

Because of the current legal hassle, we did *not* include either 386BSD or NET/2. We hope to include them on a future issue, once the dust has settled a bit.

Price: \$60, plus shipping, handling, and applicable taxes.

USENIX members may purchase the issue for \$50. Contact us for unusual cases, quantity discounts, more information and ask about the PTF Buying Plan.

The issue (two discs and a booklet) may be ordered from:

Prime Time Freeware
+1 408-738-4832 (Voice), -2050 (FAX)
415-112 N. Mary Ave., Suite 50
Sunnyvale, CA 94086 USA
[<ptf@cfcl.com>](mailto:ptf@cfcl.com)

Issue 2-1 is in production at press time. Call for details.

Calendar of Events

1993

- Feb 22-24 Sun Open Sys. Expo, Chicago, IL
Mar 8 -12 Interop, Washington, D.C.
15-19 UniForum, San Francisco, CA
31-
Apr 4 * Applications Development Symposium - POSTPONED ^
19-21 * Mach III, Santa Fe, NM
19-21 * SANS II - Washington, DC
19-23 IEEE 1003
May 20-22 UniForum NZ, New Zealand
May 25-27 NeXTWORLD, San Francisco, CA
Jun 5-11 DECUS, Atlanta, GA
21-25 * USENIX, Cincinnati, OH
Jul 12-16 IEEE 1003
Aug 1 ACM Siggraph, Anaheim, CA
2 - 3 * Mobile & Location Independent Computing, Cambridge, MA
23-27 Interop, San Francisco, CA
" INET '93, San Francisco, CA
Sept 20-22 *Microkernels II, San Deigo, CA
23-24 * SEDMS IV, San Diego, CA
Oct 4-6 * UNIX Security Symposium IV, Santa Clara, CA
18-22 IEEE 1003
Nov 1- 5 * LISA VII, Monterey, CA
Dec 4-10 DECUS, San Francisco, CA

1994

- Jan 17-21 * USENIX, San Francisco, CA
Mar 23-25 UniForum, San Francisco, CA
Spring * C++ Conference
May 7-13 DECUS, New Orleans, LA
Jun 6-10 * USENIX, Boston, MA
Sep 12-16 Interop, San Francisco, CA
Nov 12-18 DECUS, Anaheim, CA

1995

- Jan 16-20 * USENIX, New Orleans, LA
Feb 21-23 UniForum, Dallas, TX
May 13-19 DECUS, New Orleans, LA
Jun 19-22 * USENIX, San Francisco, CA
Nov 2- 8 DECUS, San Francisco, CA

1996

- Jan 22-26 * USENIX, San Diego, CA
Mar 12-14 UniForum, San Francisco, CA
May 18-24 DECUS, Orlando, FL
Nov 16-22 DECUS, Anaheim, CA

This is a combined calendar of planned conferences, symposia, and standards meetings related to the UNIX operating system. If you have a UNIX-related event that you wish to publicize, please contact login@usenix.org. Please provide your information in the same format as above.

* = events sponsored by the USENIX Association.

^ = has been postponed due to insufficient number of submissions received by the program committee.

ACM: Association for Computing Machinery
AUUG: Australian UNIX Users Group
DECUS: Digital Equipment Computer Users Society
EurOpen: European Forum for Open Systems
IEEE: Institute of Electrical and Electronics Engineers
IETF: Internet Engineering Task Force

INET: Internet Society
Interex: Intl Assoc.- Hewlett-Packard Comp. Users
JUS: Japan UNIX Society

LISA: USENIX Systems Administration Conference
SANS: Conf. on Tools & Techniques for System Admin., Networking & Security

SEDMS: Symposium on Experiences with Distributed and Multiprocessor Systems
UKUUG: United Kingdom UNIX Systems Users Group
UniForum: International Association of UNIX and Open Systems Professionals

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710

Second Class Postage
PAID
At Berkeley, California
and Additional Offices

POSTMASTER: Send address changes to *login*., USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710



What's Inside?

Microkernels Call for Papers

SAGE Election Results

POSIX - Caving In

The Bookworm